PHITS

Ver. 2.52

User's Manual

English version

Contents

1 Introduction

Particle and heavy ion transport code is an essential implement in design and study of spacecrafts and accelerator facilities. We have therefore developed the multi-purpose Monte Carlo Particle and Heavy Ion Transport code System, PHITS ,[1, 2] based on the NMTC/JAM.[3] The physical processes which we should deal with in a multipurpose simulation code can be divided into two categories, transport process and collision process. In the transport process, PHITS can simulate a motion under external fields such as magnetic and gravity. Without the external fields, neutral particles move along a straight trajectory with constant energy up to the next collision point. However, charged particles and heavy ions interact many times with electrons in the material losing energy and changing direction. PHITS treats ionization processes not as collision but as a transport process under an external field. The average dE/dx is given by the charge density of the material and the momentum of the particle taking into account the fluctuations of the energy loss and the angular deviation. The second category of the physical processes is the collision with the nucleus in the material. In addition to the collision, we consider the decay of the particle as a process in this category. The total reaction cross section, or the life time of the particle is an essential quantity in the determination of the mean free path of the transport particle. According to the mean free path, PHITS chooses the next collision point using the Monte Carlo method. To generate the secondary particles of the collision, we need the information on the final states of the collision. For neutron induced reactions in low energy region, PHITS employs the cross sections from Evaluated Nuclear Data libraries. For high energy neutrons and other particles, we have incorporated two models, JAM[4] and JQMD[5] to simulate the particle induced reactions up to 200 GeV and the nucleus-nucleus collisions, respectively.

Recently PHITS introduces an event generator for particle transport parts in the low energy region. Thus, PHITS was completely rewritten for the introduction of the event generator for neutron-induced reactions in energy region less than 20 MeV. Furthermore, several new tallis were incorporated for estimation of the relative biological effects. This report includes descriptions on new features and functions introduced into the code. For examples, GG geometry, parallelization, DPA tally, neutron, photon and electron transportation, and detailed descriptions how to setup the geometry as well. In order to keep comprehensive descriptions as the manual of PHITS , this report includes description on some parts of the NMTC/JAM code, which is an origin of code structure of PHITS .

1.1 Recent Improvements

Essences of improvements after version 2.24 are described below.

From ver. 2.26, we added the function to generate knocked-out electrons so-called δ-rays produced along the trajectory of charged particle. Setting the threshold energy for each region in the `[delta ray]` section, you can explicitly transport δ-rays above the threshold energy.

From ver. 2.28, you can use options of `dumpall` and `dump` for `[t-cross]`, `[t-time]`, and `[t-product]` tallies also on the MPI parallel computing. When these options are used in parallel computing, PHITS makes (PE−1) files for writing the dump information from each node, where PE is the total number of used Processor Elements. PHITS can also read the dump files in the parallel computing.

From ver. 2.30, the radiation damage model for calculating DPA (Displacement Per Atom) in PHITS was improved using the screened Coulomb scattering. We also added the `[multiplier]` section to be used in the `[t-track]` section.

From ver. 2.50, the following functions are implemented.

- The procedure for calculating statistical uncertainties was revised. The function to restart the PHITS calculation based the tally results obtained by past PHITS simulations was implemented in order to increase the history number when the number is not enough. Please see Sec. 4.2.2 in more detail. This improvement was performed by Mr. Daichi Obinata of Fujitsu Systems East Limited, and was supported by Center for Computational Science & e-Systems, Japan Atomic Energy Agency (JAEA).

- The shared memory parallel computing using OpenMP architecture became available in PHITS , though some restrictions still remain (see Sec. 11.2). For this purpose, we drastically revised the source code of PHITS , and old FORTRAN compilers such as f77 and g77 cannot be used for compiling PHITS anymore. See Sec. 2.4 in detail. This work was supported by Next-Generation Integrated Simulation of Living Matter, Strategic Programs for R&D of RIKEN, and RIKEN Special Postdoctoral Researchers (SPDR) Program. For this improvement, we used K computer and RIKEN Integrated Cluster of Clusters (RICC).

- The cross section data for photo-nuclear reaction was revised based on JENDL Photonuclear Data File 2004 (JENDL/PD-2004). It should be noted that the current version of *PHITS* can handle only giant resonances among the photo-nuclear reaction mechanisms. Therefore, the accuracy for calculating higher energy photo-nuclear reactions above 20 MeV is not good.

- The Statistical Multi-fragmentation Model (SMM) was implemented in the statistical decay of highly-excited residual nuclei. Owing to this implementation, the accuracy of calculating the production cross sections of light and medium-heavy fragments in heavy ion collisions was improved.

- Intra-Nuclear Cascade of Liège (INCL) was implemented, and employed as the default model for simulating nuclear reactions induced by neutrons, protons, pions, deuterons, tritons, ^{3}He and α particles at intermediate energies. This improvement was supported by Dr. Joseph Cugnon of University of Liège and Dr. Davide Mancusi, Dr. Alain Boudard, Dr. Jean-Christophe David, and Dr. Sylvie Leray of CEA/Saclay under collaboration between CEA/Saclay and JAEA.

- KUROTAMA model, which gives reaction cross sections of nucleon-nucleus and nucleus-nucleus, was implemented. This improvement was supported by Dr. Akihisa Kohama of RIKEN, Dr. Kei Iida of Kochi University, and Dr. Kazuhiro Oyamatsu of Aichi Shukutoku University.

- Intra-Nuclear Cascade with Emission of Light Fragment (INC-ELF) was implemented. Uozumi research group performed this development under collaboration between Kyushu University and JAEA.

- A user-defined tally named `[t-userdefined]` was introduced in order to deduce user specific quantities from the *PHITS* simulation. Re-compile of *PHITS* is required to use this tally. See Sec. 6.14 in detail.

- The neutron Kerma factors for several nuclei such as ^{35}Cl were revised. The photo- and electro-atomic data libraries were newly developed based on JENDL-4.0 and the Livermore Evaluated Electron Data Library (EEDL), respectively.

From ver. 2.52, the following functions are implemented.

- Electron, positron and photon transport algorithms were revised. In the new version, effective stopping powers of electrons and positions vary with their cut-off energies. The energies are conserved in an event induced by photon-atomic interactions such as the photo-electric effect.

- A new tally `[t-dchain]` was implemented to generate input files of DCHAIN-SP, which can calculate the time dependence of activation during and after irradiations. Please see Sec. 6.13 in detail.

- Macro bodies of Right Elliptical Cylinder (REC), Truncated Right-angle Cone (TRC), Ellipsoid (ELL), and Wedge (WED) are implemented.

1.2 Development members

Koji Niita,
Research Organization for Information Science & Technology (RIST).

Norihiro Matsuda, Shintaro Hashimoto, Yosuke Iwamoto, Tatsuhiko Sato, Shusaku Noda, Tatsuhiko Ogawa,
Hiroshi Nakashima, Tokio Fukahori, Keisuke Okumura, and Tetsuya Kai,
Japan Atomic Energy Agency (JAEA).

Hiroshi Iwase,
High Energy Accelerator Research Organization (KEK).

Takuya Furuta,
RIKEN.

Satoshi Chiba,
Tokyo Institute of Technology (TITech).

Lembit Sihver,
Chalmers University, Sweden.

The following members also contributed to the development of *PHITS* .

Hiroshi Takada, Shin-ichro Meigo, Makoto Teshigawara, Fujio Maekawa, Masahide Harada, Yujiro Ikeda, and Yukio Sakamoto,
Japan Atomic Energy Agency (JAEA).

Takashi Nakamura,
Tohoku University.

Davide Mancusi,
Chalmers University, Sweden.

2 Installation, compilation and execution of PHITS

The source code of *PHITS* is written in FORTRAN, and can be compiled and executed on various operating system, such as Windows, Mac and Linux. For Windows and Mac, the executable file compiled by Intel Fortran was included in the *PHITS* package. Thus, you can execute *PHITS* in Windows and Mac without compiling it. For Linux, you must compile *PHITS* by yourself using `make` command coupled with an appropriate FORTRAN compiler.

2.1 Operating environment

PHITS can be executed on Windows (XP or later), Mac (OS X v10.5 or later), Linux, and Unix operating systems. Recommended system requirements for *PHITS* are 2GB of RAM and 6GB (at least 3GB is required) of available space on hard disk.

There is no software you have to pre-install before using *PHITS* . However, we recommend you to install a text editor that can display line numbers, since line number is specified if you make a mistake in your *PHITS* input file. Furthermore, installation of Ghostscript and GSview is required to display image files created by *PHITS* , which are written in the EPS format. An example of free text editor for Windows is

- Crimson Editor (http://www.crimsoneditor.com/).

For details of the installation of Ghostscript and GSview, see the following web pages.

- Ghostscript (http://www.ghostscript.com/)
- GSview (http://pages.cs.wisc.edu/ ghost/gsview/index.htm)

You have to recompile the source code in order to extend memory usage of *PHITS* (see Sec. 2.9) or define an original radiation source using usrsors.f (see Sec. 4.3.13). Our recommended FORTRAN compilers are Intel Fortran Compiler 11.1 (or later) and gfortran 4.71 (or later). If you use other compilers, errors may occur in compiling or executing *PHITS* .

2.2 Installation and execution on Windows

For Windows, you can install and execute *PHITS* in the following way.

(1) If you have installed a previous version of *PHITS* , rename the installed folder to “phits-old” or similar.

(2) Double click “setup-eng.vbs” .

(3) Define install folder (We recommend to select “c:\”).

(4) Right click “\phits\lecture\lec01\lec01.in” and select “send to” → “phits” .

(5) Check whether “xz_track_all.eps” is created or not.

If you want to execute *PHITS* in the memory-shared parallel mode, you have to change the environmental variable “PHITS_PARALLEL” to the number of cores you want to use. You can specify this variable in “phits.bat” in the “\phits\bin” folder.

The installer “setup-eng.vbs” performs the following processes.

(1) Extract “phits.zip” into the specified installation folder.

(2) Add “\phits\bin\” in the environment variable “PATH” .

(3) Make shortcuts of three batch files, “phits.bat” and “angel.bat” in “\phits\bin” folder and “dchain.bat” in “\phits\dchain-sp\bin” folder, in “sendto” folder.

(4) Revise the first line of the nuclear data list file “xsdir.jnd” in the “\phits\data” folder as `datapath`=‘the installation folder’+ ‘\phits\XS’.

2.3 Installation and execution on Mac

Installation

Double-click “phits_installer” included in the “Mac” folder of the DVD or USB flash drive, and specify an installation folder for *PHITS* . Then, “phits” folder, which includes all contents of *PHITS* such as executable files, source code, documents for lecture, and sample input files of *PHITS* , is created in the specified folder.

(Note #1) You cannot change the folder name after installation. If you want to change the name, you have to install again.

(Note #2) In the case that the “phits” folder already exists in the specified installation folder, it would be renamed “phits[today’s date].[current time]” .

Execution

You can execute *PHITS* by drag and drop an input file onto a blue icon of *PHITS* on Dock. Results of *PHITS* are created in the folder containing the input file.

Other usage

When you use *PHITS* with “Terminal.app” , you have to type as follows:

```
echo 'export PATH=/PATH-TO-PHITS/phits/bin:${PATH}' >> ~/.bash_profile
source ~/.bash_profile
```

where “/PATH-TO-PHITS” should be changed your installation folder (e.g. /Users/noda). Then, you can execute *PHITS* with a command as below:

```
phits250_mac.exe < your_input
```

2.4 Compilation using “make” command

You can compile the *PHITS* code using “make” command. For this purpose, the “makefile” file in “src” folder should be revised to be suitable for your own computer. If you want to execute *PHITS* on a parallel computing using MPI and OpenMP, you have to set “MPIFLGS” in the 6th line and “OMPFLAGS” in the 7th line, respectively, to be “true” . Furthermore, computer system and compiler you use should be selected from candidates shown after the 13th line. Since compiler options written in “makefile” are just examples, and you may have to change them to be suitable for your computer environment.

For Windows, you have to install “make” command by yourself. It can be downloaded from the web site below.

- Make for Windows (http://gnuwin32.sourceforge.net/packages/make.htm)

If you use option ‘-j’ in the “make” command, you will get many errors because the order of compiling files should not be changed in the make of *PHITS* . (‘-j’ option changes the order). In that case, you have to type ‘make -j’ again, or do not use ‘-j’ option, then you will succeed in making the *PHITS* executable.

In order to make *PHITS* applicable to the memory-shared computing, the source code of *PHITS* was dramatically revised from version 2.50. The status of most variables used in *PHITS* was changed from “static” to “dynamic” . Consequently, *PHITS* 2.50 or later cannot be compiled by old Fortran compilers such as f77 and g77. Therefore, Fortran compilers recommended by *PHITS* office are Intel Fortran Compiler 11.1 (or later) and gfortran 4.71 (or later).

2.5 Compilation using Microsoft Visual Studio with Intel Fortran for Windows

In "\phits\bin" folder, a solution file ("bin.sln") and a project file ("phits-intel.proj") are included, which are required for compiling *PHITS* using Microsoft Visual Studio coupled with Intel Fortran. You can compile *PHITS* using these files as follows.

(1) Double-click the "bin.sln" file. (This file may be automatically updated when you use a new version of Visual Studio or Intel Fortran. You cannot open these files if you use an older version of Visual Studio (before 2005) and/or Intel Fortran (before 11.1).

(2) Build "phits-intel.vfproj" in the release mode.

(3) Make an input file for *PHITS* in the "bin" folder.

(4) Execute the project in the release mode.

(5) Type 'file=input file name' in the console window.

(6) Check whether "xz_track_all.eps" is created or not.

If you want to compile *PHITS* in the memory-shared parallel mode, you have to change "a-angel.f" to "a-angel-winopenmp.f" in "Source files" of "phits-intel.vfproj" , and add '/Qopenmp' in the additional option window (see project → property → Fortran → command line), before building "phits-intel.proj" .

If you want to use your compiled *PHITS* using the "sendto" command, please rewrite the environmental variable "PHITS_EXE" written in "phits.bat" , e.g.

```
set PHITS_EXE=C:\phits\bin\Release\phits-intel.exe
```

2.6 Compilation of ANGEL

ANGEL is a computer program for making graphs in the EPS (Enhanced PostScript) format using simple input files. Namely, *ANGEL* converts a file written by *ANGEL* computer language, which consists the minimum order to make a figure from numerical data, to that by PostScript one, which is a page description language created by Adobe Systems.

ANGEL is included in the *PHITS* sources, in other words, *ANGEL* is installed automatically in the *PHITS* code. But you will need a stand-alone *ANGEL* for off line plotting. You can compile the stand-alone *ANGEL* easily using the "make.ang" file, which is included in the *PHITS* source files. After modify the "make.ang" , execute 'make -f make.ang' to compile the stand-alone *ANGEL* . Please see *ANGEL* manual in more detail.

2.7 Executable file

PHITS code can be executed on the UNIX system by the following command,

List 3.1 ● command line to execute *PHITS*

```
phits100 < input.dat > output.dat
```

where `phits100` is the *PHITS* executable file, `input.dat` is the input file of the *PHITS* calculation, and `output.dat` is the output file for summary and error messages.

You can use the same way on the Windows system unless a parameter `infl` is used. When you use other files for the input data with this parameter, the following text should be written in the first line of `input.dat`.

List 3.2 ● the first line of the standard input

```
file = input.dat
```

This method can be used on the other systems including the UNIX. See section 3.3 for the `infl` parameter.

If you run the *PHITS* code by the parallel computing, the method shown in List 3.1 can not be used even on the UNIX system. Instead you can use the List 3.2 method on the parallel calculation. In addition, *PHITS* is forced to read the input file named `phits.in` on the parallel computing.

2.8 Terminating the PHITS code

Once *PHITS* is executed, it creates the `batch.now` file. The `batch.now` file contains an elapse information after every batch. It also contains each PE status on the parallel calculation. You can check if PE abort is occurred by the `batch.now`.

The first line of the `batch.now` is written as

```
1 <--- 1:continue, 0:stop
```

If you change the value "1" into "0", the calculation will be terminated and the summary and results are provided for the histories until terminated. It is an useful function shown in below.

Associated with the batch.now, a new parameter was included by the parameter section. You can specify it as

```
itall = 2  # (D=0) 0:no tally at batch, 1:same, 2:different
```

If you set `itall = 1`, *PHITS* outputs the latest results (tally output) after every batch. On the parallel calculation, results are created by every batch $\times$ (PE -1). In the case of `itall = 1`, the results are overwritten in same files. On the other case of `itall = 2`, the results after every batch are written in different named file. Results are named by user specified name + batch number. The final results are written in the user specified file.

By using this functions, you can terminate a *PHITS* calculation at any time with checking a latest result. Also you can monitor the latest results with graphical plot automatically made by *PHITS* (See section 5.7.16).

`rijk` written in the `batch.now` file is the initial random number of the current batch. For example, in cases of unsuccessful termination of *PHITS* , you can reproduce the calculation of the specified batch using the value of `rijk`.

2.9 Array sizes

You should check and modify the array sizes described in the param.inc file. The "mdas" is the most important variable. It specifies the total size of arrays for geometry, tally output, nuclear data, and bank. You can find out the current use in a input echo (corresponds " output.dat" in previous example).

The bank size can be set in the parameter section. If the bank becomes full, odd arrays in mdas are used. The default param.inc is shown below.

List 3.3 ● param.inc

```
 1:  ************************************************************************
 2:  *                                                                      *
 3:  *     'param.inc'                                                      *
 4:  *                                                                      *
 5:  ************************************************************************
 6:
 7:        parameter ( mdas   =20000000 )
 8:        parameter ( kvlmax =    3000 )
 9:        parameter ( kvmmax = 1000000 )
10:        parameter ( itlmax =      60 )
11:        parameter ( inevt  =      70 )
12:        parameter ( isrc   =      50 )
13:        parameter ( latmax = 2000000 )
14:
15:        common /mdasa/ das( mdas )
16:        common /mdasb/ mmmax
17:
18:  *----------------------------------------------------------------------*
19:  *                                                                      *
20:  *       mdas   : total memory * 8 = byte                               *
21:  *       mmmax  : maximum number of total array                         *
22:  *                                                                      *
23:  *       kvlmax : maximum number of regions, cell and material          *
24:  *       kvmmax : maximum number of id for regions, cel and material    *
25:  *                                                                      *
```

```
26:  *       itlmax : number of maximum tally entry                         *
27:  *       inevt  : number of collision type for summary                  *
28:  *       isrc   : number of multi-source                                *
29:  *       latmax : maximum number of lattice in a cell                   *
30:  *                                                                      *
31:  *----------------------------------------------------------------------*
```

3 Input File

PHITS input consists of some sections as listed in Table 3.1 and 3.2. Each sections begins from a `[Section Name]`. You can put maximum 4 blanks between the line head and the declaration of `[Section Name]`, otherwise (more than 4 blanks) `[Section Name]` is not recognized as a beginning of a section and the following part is regarded as items of the previous section.

3.1 Sections

Table 3.1 and 3.2 shows the various sections used in *PHITS* .

Table 3.1: Sections(1)

name	description
[title]	Title
[parameters]	Various type of parameters
[source]	Source definition
[material]	Material definition
[body]	CG body definition
[region]	Region definition by CG
[surface]	Surface definition by GG
[cell]	Cell definition by GG
[transform]	Definition the coordinate transform of GG surface
[importance]	Region importance definition
[weight window]	Weight window definition
[volume]	Region volume definition
[temperature]	Cell temperature definition
[brems bias]	Bremsstrahlung bias definition
[photon weight]	Photon product weight definition
[forced collisions]	Forced collision definition
[magnetic field]	Magnetic field definition
[counter]	Counter definition
[reg name]	Region name definition for graphical plot
[mat name color]	Material name and color definition for graphical plot
[mat time change]	time dependent material definition
[super mirror]	super mirror definition
[elastic option]	elastic option definition
[timer]	timer definition
[delta ray]	production of δ-rays
[multiplier]	multiplier definition

Table 3.2: Sections(2)

name	description
[t-track]	Track length tally definition
[t-cross]	Surface crossing tally definition
[t-heat]	Heat developing tally definition
[t-deposit]	Deposit tally definition
[t-deposit2]	Deposit2 tally definition
[t-yield]	Residual nuclei yield tally definition
[t-product]	Produced particle tally definition
[t-dpa]	DPA tally definition
[t-let]	LET tally definition
[t-sed]	SED tally definition
[t-time]	Time tally definition
[t-star]	Star density tally definition
[t-dchain]	Dchain tally definition
[t-userdefined]	User defined tally definition
[t-gshow]	Region surface display definition for graphical plot
[t-rshow]	Physical quantity region display definition for graphical plot
[t-3dshow]	3D graphical geometry plot definition
[end]	End of input file

It is noted that *PHITS* does not read any input informations which are written below the [end] section.

3.2 Reading control

(1) Uppercase, lowercase, blank
Discrimination between lowercase and uppercase characters is not performed in the *PHITS* input except for file names. Blanks at line head and end are taken no account except for the declaration of the [Section Name] as described before.

(2) Tab
A tab is replaced into 8 blanks.

(3) Line Connecting
If you add " \ " at line end, the next line is considered to be a continuation line. You can use multiple lines to write input data by the " \ " connecting.

But you don't need to use the " \ " connecting in the def of [body] and the def of [region]. In these area, line is connected automatically without any symbol. This function can be also used in the [cell] and [surface] sections but in these case, more than 4 blanks are required at the line head of connected line. Details of this function are explained later.

(4) Line dividing
Short lines can be displayed in a line by dividing " ; " as

```
idbg = 0 ;  ibod = 1 ;  naz =  0
```

But this function is not available where the format is defined such as in the mesh description.

(5) Comment marks
You can use the following comment marks "#", "%", "!", "$". The comment out is effective from the comment mark to the line end. You can also use "c" as a comment mark if you put "c" in the first 5 column at line head with blanks as "c ". In the [cell] and [surface] sections, "#" is used for cell definitions, so only "$" is available as the comment mark in these [cell] and [surface] sections.

(6) Blank lines
Blank lines, and lines which begins from a comment mark are skipped.

(7) Section reading skip
If you add "`off`" after a section name as "`[Section Name] off`" the section is skipped (is not read).
As described later, you can write any comments after "`[body]`" section name as "`[body] HIMAC_experiment`"
But you can not use comments which begins `off` here, if not, the `[body]` section is skipped.

(8) Skip in sections
You can skip from any place in sections by putting `qp:` at the line head. Lines from `qp:` to the end of the section are skipped.

(9) Skip all
`q:` can be used as a terminator of a input file. It works same as the `[end]`.

3.3 Inserting files

You can include other files in any place by

`infl:` { *file.name* } [$n_1 - n_2$]

You should specify a name of a file to be inserted in { }, and the number of lines from n_1 to n_2 of that file in [ ]. If there is no [ ], PHITS includes all lines of the specified file. You can use following style to specify line numbers,
[n_1-],
[$-n_2$].
From line number n_1 to the end, and from top to line number n_2 respectively. The file insertion can be nested more than once. The including file can be nested more than once. If you write any comments in the line, square-bracket characters "[" and "]" cannot be used.

When you use the command-line interpreter (Command prompt) on the Windows system for executing PHITS , you have to be careful. If `infl` is used, you should write the following text in the first line of the input file.

```
file = input.dat
```

Here, `input.dat` is the input file name. See section 2.7.

3.4 User definition constant

You can set your own constant as

`set:` c1[*52.3*] c2[*2* * pi] c3[c1 * *1.e-8*]

This "set" definition can be written in anywhere. Defined user-constants can be used as numerical values in your input file. User-constants can be re-defined any time, and these values are kept until re-defined. In the 3rd case of above example (`c3`), another user-constant `c1` is called in a user-constant definition. In the case, the value in which the user-constant `c1` keeps at that time, is used. So even if you re-define the `c1` below the `c3` definition, the value of `c3` defined here is not changed. `pi` is set to the value of π by default.

3.5 Using mathematical expressions

Mathematical expressions can be used in your input file. It is FORTRAN style. Available functions are shown in Table 3.3.

For example,

Table 3.3: Intrinsic Function.

Intrinsic Function							
FLOAT	INT	ABS	EXP	LOG	LOG10	MAX	MIN
MOD	NINT	SIGN	SQRT	ACOS	ASIN	ATAN	ATAN2
COS	COSH	SIN	SINH	TAN	TANH		

```
param = c1 * 3.5 * sin( 55 * pi / 180 )
```

As above example as a single numerical value is expected after `param =`, you can put blanks in the expressions. However it is not allowed that multiple numerical values are aligned, such as in the `[body]` and `[region]` sections. In such region, you can close the expressions using { }, like { `c1 * 2 / pi` }.

3.6 Using the CG or GG

You have to choose between the CG system and the GG system in order to describe a geometry of calculation. When you choose the CG, you must use the `[region]` and `[body]` sections. Or GG, you must use the `[cell]`, `[surface]`, and `[transform]` sections.

You can not call both CG and GG systems at the same time. You can use the section-off feature as "`[region] off`", if CG and GG descriptions exist together in a input file.

3.7 Particle identification

Available particles in *PHITS* are identified as in Table 3.4. These particles can be specified by the symbol or the kf-code. The particles which is not specified the symbol in Table 3.4, are specified by only kf-code.

The other particles identified as type 11 can be defined by the kf-code as shown in followings, and these decay-channels and life-times are also shown in below.

By adopting the QMD code, nucleus can be treated in *PHITS* . The writing form of nuclide is as 208Pb, 56Fe. The writing style Pb, Fe, etc., means all isotopes (This can not be used as projectile). Nucleus can be described by kf=Z * 1000000 + A for the kf-code.

In the previous version of *PHITS* , the photon was called “gamma” but it is called “photon” in the newer version.

Table 3.4: List of the transport particles.

ityp	symbol	kf-code	particle name
1	proton	2212	proton
2	neutron	2112	neutron
3	pion+	211	π^+
4	pion0	111	π^0
5	pion−	−211	π^-
6	muon+	−13	μ^+
7	muon−	13	μ^-
8	kaon+	321	K^+
9	kaon0	311	K^0
10	kaon−	−321	K^-
11	other	below	other particle
12	electron	11	e^-
13	positron	−11	e^+
14	photon	22	γ
15	deuteron	1000002	deuteron
16	triton	1000003	triton
17	3he	2000003	^{3}He
18	alpha	2000004	α
19	nucleus	Z*1000000+A	nucleus
20	all	–	all particles

ityp	symbol	kf-code	particle name
11	–	+ − 12	$\nu_e\ \overline{\nu_e}$
11	–	+ − 14	$\nu_\mu\ \overline{\nu_\mu}$
11	–	−2212	$\overline{\mathrm{p}}$
11	–	−2112	$\overline{\mathrm{n}}$
11	–	−311	$\overline{\mathrm{K}^0}$
11	–	+ − 221	$\eta\ \overline{\eta}$
11	–	331	η'
11	–	+ − 3122	$\Lambda^0\ \overline{\Lambda^0}$
11	–	+ − 3222	$\Sigma^+\ \overline{\Sigma^+}$
11	–	+ − 3212	$\Sigma^0\ \overline{\Sigma^0}$
11	–	+ − 3112	$\Sigma^-\ \overline{\Sigma^-}$
11	–	+ − 3322	$\Xi^0\ \overline{\Xi^0}$
11	–	+ − 3312	$\Xi^-\ \overline{\Xi^-}$
11	–	+ − 3334	$\Omega^-\ \overline{\Omega^-}$

Table 3.5: Decay channel and life time

								blanking fraction	life time (sec)
π^0	$\rightarrow$	γ	+	γ				100%	0
π^+	$\rightarrow$	μ^+	+	ν_μ				100%	2.6029e−8
π^-	$\rightarrow$	μ^-	+	ν_μ				100%	2.6029e−8
μ^+	$\rightarrow$	e^+	+	$\overline{\nu_e}$	+	ν_μ		100%	2.19703e−6
μ^-	$\rightarrow$	e^-	+	$\overline{\nu_e}$	+	ν_μ		100%	2.19703e−6
K^0	$\rightarrow$	π^+	+	π^-				68.61%	8.922e−11
	$\rightarrow$	π^0	+	π^0				31.39%	
	$\rightarrow$	γ	+	γ				other	
K^+	$\rightarrow$	μ^+	+	ν_μ				63.51%	1.2371e−8
	$\rightarrow$	π^+	+	π^-				other	
K^-	$\rightarrow$	μ^-	+	ν_μ				63.51%	1.2371e−8
	$\rightarrow$	π^+	+	π^-				other	
η	$\rightarrow$	γ	+	γ				38.9%	0
	$\rightarrow$	π^0	+	π^0	+	π^0		31.9%	
	$\rightarrow$	π^+	+	π^-	+	π^0		23.7%	
	$\rightarrow$	π^+	+	π^-	+	γ		other	
η'	$\rightarrow$	π^+	+	π^-	+	η		44.1%	0
	$\rightarrow$	π^0	+	π^0	+	η		20.5%	
	$\rightarrow$	π^+	+	π^-	+	γ		30.1%	
	$\rightarrow$	γ	+	γ				other	
Λ^0	$\rightarrow$	p	+	π^-				64.1%	2.631e−10
	$\rightarrow$	n	+	π^0				other	
Σ^+	$\rightarrow$	p	+	π^0				51.57%	7.99e−11
	$\rightarrow$	n	+	π^+				other	
Σ^0	$\rightarrow$	Λ^0	+	γ				100%	0
Σ^-	$\rightarrow$	n	+	π^-				100%	1.479e−10
Ξ^0	$\rightarrow$	Λ^0	+	π^0				100%	2.90e−10
Ξ^-	$\rightarrow$	Λ^0	+	π^-				100%	1.639e−10
Ω^-	$\rightarrow$	Λ^0	+	K^-				67.8%	8.22e−11
	$\rightarrow$	Ξ^0	+	π^-				23.6%	
	$\rightarrow$	Ξ^-	+	π^0				other	

4 Sections format

4.1 [T i t l e] section

In the section, you can define a title of your calculation. Any numbers of title lines are allowed. Blank lines are skipped in this section.

```
[ T i t l e ]
This is a test calculation of PHITS.
Any number of title lines are allowed.
..........
```

4.2 [P a r a m e t e r s] section

The various parameters of *PHITS* can be defined in this section. The format is as below.

```
[ P a r a m e t e r s ]
para1 = number | file.name
para2 = number | file.name
..........
```

You can change the order of parameters. Each parameter has the default value. So undefined parameters use the default values.

Parameters and default values are shown in followings. (D=) means the default value.

4.2.1 Calculation mode

Table 4.1: parameter 1

parameter	value	explanation
icntl	(D=0)	basic control option
	= 0	normal *PHITS* calculation
	= 1	nuclear reaction calculation, (under development)
	= 2	output a CGVIEW input file
	= 3	output only input echo for checking memory usage, and library, and file links
	= 4	output a MARS-PF input file
	= 5	no reaction, no ionization. all regions are made be void for geometry check, and volume and area calculations
	= 6	source check, source particles can be tallied by [t-product]
	= 7	execute [t-gshow] tally (graphical output)
	= 8	geometry output of xyz mesh tally with gshow option (graphical output)
	= 9	execute [t-rshow] tally (graphical output)
	= 10	geometry output of reg mesh tally with rshow option (graphical output)
	= 11	execute [t-3dshow] tally (graphical output)
	= 12	re-calculate from dumpall file, dumpall file is specified by file(15)

The function of nuclear reaction calculation specified as icntl=1 is under development.

By setting icntl=12, *PHITS* re-calculates whole transport by reading the information from dumpall file, which is created by dumpall=1 option. The re-calculation can describe whole transport events which were calculated before. One needs the same input file as used in the previous calculation. maxcas and maxbch cannot be changed, but are read from the file. It is very powerful when you want to calculate different tallies which are not used in the previous calculation. However, please be careful that the calculation with dumpall=1 may create huge dumpall file. This option is only available for GG geometry.

4.2.2 Number of history and Bank

Table 4.2: parameter 2

parameter	value	explanation
irskip	(D=0)	random number control
	irskip>0	begin calculation after skipping histories by number of irskip (for debug)
	irskip<0	begin calculation after skipping random numbers by number of irskip (for manual parallel computing)
rseed	(D=0.0)	initial random number option
	rseed<0	get a initial random number from starting time.
	rseed=0	6647299061401 (default)
	rseed>0	use rseed as initial seed of random number
maxcas	(D=10)	Number of history per 1 batch. The upper limit is 2147483647.
maxbch	(D=10)	Number of batch. The upper limit is 2147483647.
maxbnk	(D=10000)	size of bank array
istdev	(D=0)	Control parameter for calculation procedure of statistical uncertainties and the restart mode.
	=-2	The restart calculation mode is activated, but if there is no past tally result, a new calculation is started with istdev=2.
	=-1	The restart calculation mode is activated, but if there is no past tally result, a new calculation is started with istdev=1.
	=0	istdev is automatically set to 1 for memory-shared parallel computing, and 2 for the other cases.
	=1	A new calculation is started. Statistical uncertainties are estimated from the variances of each batch result.
	=2	A new calculation is started. Statistical uncertainties are estimated from the variances of each history result. You cannot select this option in the memory-shared parallel mode.

In the *PHITS* parallel calculation, number of batch should be an integer multiple of (all number of PE −1). If not, *PHITS* converts automatically the batch number as it becomes an integer multiple and as the total number of history becomes almost same with given total histories. In this case, some comments are output at the end of an input echo.

The procedure for calculating statistical uncertainties was revised from version 2.50. The function to restart the *PHITS* calculation based on the tally results obtained by past *PHITS* simulations was implemented in order to increase the history number when the number is not enough. In this mode, the initial random seed is also read from the past tally file.

In the calculation of statistical uncertainties, you can select two modes, “batch variance mode” and “history variance mode” , which calculate the standard deviations using variances between tally results of each batch and history, respectively. In both modes, standard deviation σ is calculated by

$$\sigma = \sqrt{\frac{\sum_{l=1}^{N}(x_i w_i/\bar{w})^2 - N\bar{X}^2}{N(N-1)}} \tag{1}$$

where N is the total batch number (istdev=1) or the total history number (istdev=2), x_i and w_i are tally results and the source weight of each sample, respectively, and $\bar{X}$ and $\bar{w}$ are the mean values of the tally results and source weights of N samples, respectively. The ratio of σ to $\bar{X}$ is written as relative error in the tally output file. When you use the shared-memory parallel computing, you can select only the batch variance mode, since it is impossible to calculate the tallied quantities history by history using the shared-memory parallel computing. The standard deviations calculated by the batch variance mode vary with the combination of maxcas and maxbch, even for the same total history number. In principle, larger maxbch gives more reliable statistical uncertainties in the batch

variance mode, but it may take longer computational time. We recommend setting `maxbch` to more than 10 to obtain reliable results. On the other hand, in the history variance mode, the standard deviation depends only on the total history number, and they are independent of the combination of `maxcas` and `maxbch`. Therefore, the history variance mode (`istdev = 2`) is recommended to be selected except for the memory-shared parallel computing. However, the computational time occasionally becomes extremely long in the history variance mode, especially in the case of tallies using a lot of memories, e.g. xyz mesh tally with very fine structure. In the tallies for calculating the variance of deposition energies by each history such as `[t-deposit]` tally with `output=deposit` and `[t-deposit2]`, the standard errors instead of the standard deviations are outputted as the relative errors. The relative standard errors can be estimated by $1/\sqrt{K}$ where K is the number of history contributing to the result of the tally. This calculation procedure is independent of the `istdev` parameter.

Relative errors are generally outputted in the "r.err" column, which is placed in the rightmost column of the tally results. In the case of 2D-plot such as tallies with `axis=xy` or `rz`, errors are output in another file named "*_err" , where "*" indicates the file name specified in the tally. For example, when `file=tally.out`, the name of the error file is "tally_err.out" . This error file has the same format of the conventional tally output file. Hence, you can obtain a graph for the error in 2D-plot through a conversion process by *ANGEL* .

When `istdev<0`, the restart calculation mode is activated, but if there is no past tally result, a new calculation is started with `istdev = abs(istdev)`; namely the batch and history variance modes are selected for `istdev = -1` and `-2`, respectively. In the case of the restart calculation, the variance mode is automatically determined from the past *PHITS* calculation. You have to set the same tally parameters as written in the past tally results, but you can add new tallies in the restart calculation. The procedure for the restart calculation is given below:

(1) Check whether the file specified by `resfile` in each tally section exists or not. The default file name of "resfile" is that given by `file` parameter.

(2) If there is no resfile for a tally, it is regarded as new one. If "resfile" for all tallies cannot be found, a new calculation is started with `istdev = abs(istdev)`.

(3) If resfile exists, *PHITS* reads from the file about the information on the variance mode `istdev`, total weight `resc2`, total history number `resc3`, history number per batch `maxcas`, and the next random seed `rijklst`, and results and relative errors of the past calculation.

(4) Check the consistency between tally parameters given in the current and past *PHITS* input files. If they are not consistent each other, *PHITS* stops the calculation and output an error message. It should be noted that the consistencies of not all tally parameters are checked in this process.

(5) Check the consistency of `istdev` and `maxcas` (in the batch variance mode only) among the resfiles. If they are consistent, the restart calculation is performed using those valued. If the inconsistency is found, the calculation is stopped.

(6) Change the initial random seed to `rijklst` obtained from the first resfile. If `rijklst` written in resfiles are different from one another, a warning message is outputted.

(7) When the restart calculation is finished, the tally results are outputted in the file specified by `file=`. In the case that "resfile" is not specified, the past tally results are overwritten.

Important notice:

I. All past tally results should be calculated in the same variance mode: i.e. `istdev` in all "resfile" should be the same.

II. `maxcas` written in the input file is not used in the restart calculation of the batch variance mode.

III. The consistencies of input parameters except for those given in the tally sections are not checked. Hence, you have to make them identical by yourself in the restart calculation.

4.2.3 Cut off energy and switching energy

Table 4.3: parameter 3

parameter	value	explanation
emin(1)	(D=1.0)	proton cut off energy (MeV)
emin(2)	(D=1.0)	neutron cut off energy (MeV)
emin(i)	(D=1.0)	cut off energy for i-th particle (MeV)
	i = 3-10	(i; particle id, see Table 3.4)
emin(11)	(D=2.0)	cut off energy for others (MeV)
emin(i)	(D=1.e+9)	cut off energy for i-th particle (MeV)
	i = 12-19	(i; particle id, see Table 3.4)
	i = 15-19	energy unit is [MeV/nucleon]
esmin	(D=0.001)	minimum energy for range calculation for the charge particles (MeV)
esmax	(D=300000)	maximum energy for range calculation for the charge particles (MeV)
cmin(i)	(D=emin(i))	nuclear reaction cut off energy for i-th particle (MeV)
		any nuclear reactions under cmin(i) are not treated
	i = 15-19	for these nucleus, energy unit is [MeV/nucleon]
dmax(i)	(D=emin(i))	maximum energy of library use for i-th particle

A value itself given by these parameters is included in a lower limit, and not included in a upper limit. For example, a proton at just the energy of emin(1) is not cut off.

PHITS uses libraries in the energy region $emin < energy < dmax$. If you set $emin \geq dmax$, any libraries are not used. The maximum energies for proton, neutron, photon, and electron are 150 MeV, 150 MeV, 100GeV, and 1000 MeV respectively in this version.

We create the range table of charge particles in $esmin < energy < esmax$. If you want to use much larger energy, you should set *esmax*.

Table 4.4: parameter 4

parameter	value	explanation
ejamnu	(D=3500.)	switching energy of nucleon nucleus reaction calculation from Bertini (or QMD) to JAM model (MeV)
ejampi	(D=2500.)	switching energy of pion nucleus reaction calculation from Bertini to JAM model (MeV)
eisobar	(D=0.0)	maximum energy (MeV) of isobar calculation when isobar is defined (isobar=1)
eqmdnu	(D=3500.)	switching energy of nucleon nucleus reaction calculation from Bertini to QMD model (MeV)
eqmdmin	(D=10.0)	minimum energy of QMD calculation [MeV/nucleon]
ejamqmd	(D=3500.0)	switching energy from JQMD to JAMQMD [MeV/nucleon]
inclg	(D=1)	Control parameter for use of INCL.
	=0	Not use of INCL.
	=1	Use of INCL in a proton, neutron, pion, $d, t,^3$He, or α induced reaction.
	=2	Use of INCL in a proton, neutron, or pion induced reaction.
einclmin	(D=1.0)	Minimum energy of INCL calculation (MeV/u).
einclmax	(D=3000.0)	Maximum energy of INCL calculation (MeV/u).
incelf	(D=0)	Control parameter for use of INC-ELF.
	=0	Not use of INC-ELF.
	=1	Use of INC-ELF in a proton or neutron induced reaction.
eielfmin	(D=1.0)	Minimum energy of INC-ELF calculation (MeV).
eielfmax	(D=3500.0)	Maximum energy of INC-ELF calculation (MeV).

Below *eqmdmin*, we do not consider the nuclear reactions of d, t, α, and nucleus. Since the applicability of QMD is restricted in low energy region and the range of nucleus is very short in the normal material, one do not need consider the low energy reactions of nucleus for usual case. High energy heavy ion collisions are treated by JAMQMD above 3.5GeV/u in default. This switching energy can be changed by *ejamqmd*. Even for nucleon induced collisions, you can calculate the collisions by JAMQMD by changing *eqmdnu*, *ejamnu* and *ejamqmd*.

INCL(Intra-Nuclear Cascade of Liège) is a nuclear reaction model for nucleons (proton and neutron), pions, and light-ions ($d, t,^3$He, or α) induced reactions. From version 2.50, INCL is used by default for these reactions if you don't explicitly specify the nuclear reaction model. In the case of using results obtained by INCL in your publication, please refer a document[1] shown below.

INC-ELF(Intra-Nuclear Cascade with Emission of Light Fragment) is a nuclear reaction model for nucleons induced reactions. In the case of using results obtained by INC-ELF in your publication, please refer a document[2] shown below.

[1] J. Cugnon, A. Boudard, S. Leray, and D. Mancusi, J. Korean Phys. Soc. 59, 955 (2011).
[2] Y. Sawada, Y. Uozumi, S. Nogamine, T. Yamada, Y. Iwamoto, T. Sato, and K. Niita, Nucl. Instr. & Meth. B 291, 38-44 (2012).

4.2.4 Cut off time, cut off weight, and weight window

Table 4.5: parameter 5

parameter	value	explanation
tmax(i)	(D=1.e+9) i = 1-20	cut off time for i-th particle (nsec) (i; particle id, see Table 3.4)
wc1(i) wc2(i) swtm(i)	(D=-0.5) (D=wc1/2) (D=1.0)	minimum weight for i-th particle cutoff weight for i-th particle minimum source weight for i-th particle
wupn	(D=5)	maximum value of weight window = minimum value in [Weight Window] section × wupn wupn ≥ 2
wsurvn	(0.6*wupn)	survival weight value 1 < wsurvn < wupn
mxspln	(D=5)	maximum number of split, maximum multiple number of survival mxspln > 1
mwhere	(D=0)	where the weight window takes place -1: at nuclear reaction, 0:both, 1: at region crossing

Cut off time should be specified as tmax(i)= [nsec] for each particle. After elapsing the cut off time, the particle is killed. It is not effective to results for high energy particle transport, but it is useful for low energy particle transport calculation.

Weight of a particle is changed by the importance, forced collisions, implicit captures, and weight window functions. When the weight takes lower value than user-defined weight cut off, the particle is judged if it is killed or not by the Russian roulette method. This function is not available for particles defined in the weight window.

In the Russian roulette method, when the weight WGT is lower than the product of $WC2$ and ratio R of two importances between at source point and at current point, $WC2 \times R$ (i.e., when $WGT < WC2 \times R$), the particle survives with a probability, $WGT/(WC1 \times R)$, which is a function of own weight WGT. Then the weight is changed as $WGT = WC1 \times R$. Unless, the particle is killed. If the $WC1$ and $WC2$ are given as negative, $|WC1| \times SWTM$ and $|WC2| \times SWTM$ are set as $WC1$ and $WC2$.

If there are any particles and regions which are not set importance, these importances are set as 1

4.2.5 Model option (1)

Table 4.6: parameter 6

parameter	value	explanation
ielas	(D=2)	elastic scattering option
	= 0	exclude elastic scatter
	= 1	include neutron elastic scatter
	= 2	include neutron and proton elastic scatter
ielms	(D=100)	number of angle group for elastic scattering
inmed	(D=1)	nucleon-nucleon cross section options for Bertini model
	= 0	free (nmtclb25.dat)
	= 1	Cugnon old (nmtclb95.dat)
	= 2	Cugnon new (nmtclb30.dat)
nevap	(D=3)	options for Evaporation model
	= 0	without evaporation model
	= 1	using DRES model
	= 2	using SDM model
	= 3	using GEM model
ismm	(D=0)	Control parameter of Statistical Multi-fragmentation Model (SMM)
	= 0	SMM is not used.
	= 1	SMM is used. When a JQMD calculation is performed, switching time from JQMD to GEM changes from 100fm/c, which is the default value, to 75fm/c.
igamma	(D=0)	γ decay option for residual nuclei
	= 0	without γ decay
	= 1	with γ decay; file(14) is required. file(14)=trxcrd.dat
isobar	(D=0)	options for isobar model
	= 0	without isobar
	= 1	with isobar
ipreeq	(D=0)	options for pre-equilibrium model (when nevap=1)
	= 0	without pre-equilibrium model
	= 1	with pre-equilibrium model

It is noted that inmed=1 is the default value.

Using the Statistical Multi-fragmentation Model (SMM), the accuracy of calculating the production cross sections of light and medium-heavy fragments is improved in collisions of heavy ions such as Pb and Hg or for incident energies over 100 MeV/u. It should be noted that the computational time becomes long using this model.

4.2.6 Model option (2)

Table 4.7: parameter 7

parameter	value	explanation
ieleh	(D=0)	options for electron and positron transport
	= 0	no slowing down, no reaction, in the energy region above dmax(12)
	= 1	make e=dmax(12), and weight=e/dmax(12) in the energy region above dmax(12)
ipnint	(D=0)	Options for photo-nuclear reaction.
	= 0	This reaction mechanism is not taken into account.
	= 1	This reaction mechanism is taken into account. (Until ver. 2.30, this parameter was ipngdr.)
level	(D=3)	level density option when nevap=1
	= 1	8/A
	= 2	with Baba's parameters
	= 3	with Igunatyuk's parameters
npidk	(D=0)	treatment of minus charged decay particles below cut off energy
	= 0	make absorbed by force
	= 1	make decayed
imagnf	(D=0)	Magnetic field
	= 0	without Magnetic field
	= 1	with Magnetic field
andit	(D=0)	Δ angular distribution for Bertini
	= 0	50% isotropic, 50% forward
	= 1	all isotropic
	= 2	all forward
icxsni	(D=0)	Option for reaction, elastic, and total cross sections in nucleon-nucleus collisions.
	= 0	Pearlstein-Niita' formula
	= 1	KUROTAMA model
	= 2	Sato' formula
icrhi	(D=1)	option for total cross section for Nucleus-Nucleus collision
	= 0	Shen formula
	= 1	NASA formula
	= 2	KUROTAMA model

If a particle, which has decay channel, takes lower energy than cut off, the particle decays completely. In such decay particles, minus charged particles with npidk = 0, are forced to take reaction for the purpose of forced absorption. If it is not absorbed, then the particle is made decayed.

KUROTAMA model gives reaction cross sections of nucleon-nucleus and nucleus-nucleus for wide incident energy region. See the document[3] below for detail. You have to refer this document when you use the results obtained by the KUROTAMA model in your publications.

[3] K. Iida, A. Kohama, and K. Oyamatsu, J. Phys. Soc. Japan 76, 044201 (2007).

4.2.7 Model option (3)

Table 4.8: parameter 8

parameter	value	explanation
gravx	(D=0)	x-component of gravity direction
gravy	(D=0)	y-component of gravity direction
gravz	(D=0)	z-component of gravity direction
ndedx	(D=2)	option for dE/dx of charged particle and nucleus
	= 0	SPAR for nucleus, NMTC for the others
	= 1	ATIMA for nucleus and proton, NMTC for the others
	= 2	SPAR for nucleus, proton, pion, and muon, NMTC for the others
ih2o	(D=-1)	Water (only for H2O) Ionization Potential option for ATIMA
	< 0	default, 75 eV
	> 0	Ionization Potential for water(eV)
nspred	(D=0)	option for Coulomb diffusion (angle straggling)
	= 0	without Coulomb diffusion
	= 1	with original Coulomb diffusion
	= 2	with Moliere First Coulomb diffusion
	= 3	with Moliere Third Coulomb diffusion
	= 10	with Coulomb diffusion for ATIMA
nedisp	(D=0)	energy straggling option for charged particle and nucleus
	= 0	without energy straggling
	= 1	with Landau Vavilov energy straggling
	= 10	with energy straggling for ATIMA
e-mode	(D=0)	option for event generator mode
	= 0	normal mode
	= 1	event generator mode, file(14) = trxcrd.dat is required
usrmgt	(D=1)	option for user subroutine of time dependent magnetic field
	= 1	usrmgt1.f is used, which includes Wobbler magnet.
	= 2	usrmgt2.f is used, which includes Pulse magnet.
usrelst	(D=1)	option for [elastic option]
	= 1	usrelst1.f is used, which is for Bragg scattering
	= 2	usrelst2.f is used, which is a sample program

gravx, gravy, gravz represent directions of gravity. The gravity force acts on neutrons below 1 eV. For an example, for gravx=1, gravy=0, gravz=0 case, the direction of the gravity is negative direction of x-axis.

It should be noted that the default option for ndedx was ndedx=0 until *PHITS* ver. 2.00.

The option for ATIMA is under development, and it takes very long cpu time to calculate.

4.2.8 Output options (1)

Table 4.9: parameter 9

parameter	value	explanation
incut	(D=0)	neutron output options below cut off
	= 0	no output
	= 1	output in the ncut file specified as file(12)
	= 2	output in file(12) with time information
igcut	(D=0)	γ-ray and electron output options below cut off
	= 0	no output
	= 1	output γ-ray data in the gcut file specified as file(13)
	= 2	output γ-ray in file(13) with time information
	= 3	output γ-ray, electron, and positron data in file(13)
ipcut	(D=0)	proton output options below cut off
	= 0	no output
	= 1	output in the pcut file specified as file(10)
	= 2	output in file(10) with time information
inpara	(D=0)	ncut file name options in the parallel calculation
	= 0	/wk/uname/file-name of file(12)
	= 1	/wk/uname/file-name of file(12)+(PE number)
	= 3	file-name of file(12)+(PE number)
igpara	(D=0)	gcut file name options in the parallel calculation
	= 0	/wk/uname/file-name of file(13)
	= 1	/wk/uname/file-name of file(13)+(PE number)
	= 3	file-name of file(13)+(PE number)
ippara	(D=0)	pcut file name options
	= 0	/wk/uname/file-name of file(10)
	= 1	/wk/uname/file-name of file(10)+(PE number)
	= 3	file-name of file(10)+(PE number)
	/wk/uname/	/wk/ is the default directory name
		uname is a user-name read in from environment variable LOGNAME

It is noted that the default of incut and igcut were changed to be 0.

In the parallel computing, files corresponding to each PE (Processor Element) are created for writing the output. If you set inpara, igpara, or ippara=0 or 1, a file is made in the directory named by /wk/uname/ on each of the nodes. If you set inpara, igpara, or ippara=1 or 3, the each PE number is put at the end of the filename. The each PE writes down its result on only the corresponding file.

4.2.9 Output option (2)

Table 4.10: parameter 10

parameter	value	explanation
itall	(D=0)	options for tally output after every batch
	= 0	no output
	= 1	in same file
	= 2	in different files
		file name = specified file name + batch number
itstep	(D=0)	option for timing of tally for changing momentum, like magnetic field
	= 0	tally at reaction or surface cross (normal)
	= 1	tally at each step of the transport
imout	(D=0)	option for specifying the output format of [material] section
	= 0	e.g. mat[12], 208Pb.33c
	= 1	e.g. mat[12], Pb-208.33c
	= 2	e.g. m12, 82208.33c (MCNP type)
jmout	(D=0)	option for specifying the output format of material density in [material] section
	= 0	no conversion
	= 1	converting particle density
kmout	(D=0)	option of nuclear data information
	= 0	no display
	= 1	writing in input echo
matadd	(D=1)	treatment of different densities in same material in GG
	= 0	same material number
	= 1	using new material number

Normally, the tallies are called at the reaction point or at the surface crossing. Thus the particle track in the magnetic field, for an example, is shown as a straight line between collisions or between one collision and surface crossing. If you specify itstep = 1, the trajectory is described correctly as a curve. The maximum step for the magnetic field is set by deltm.

4.2.10 Output option (3)

Table 4.11: parameter 11

parameter	value	explanation
iggcm	(D=0) = 0 = 1	option of GG warnings no echo in input echo
ivout	(D=0) = 0 = 1	volume display options in the input echo in [volume] section in [region] section
ipout	(D=1) (D=0 for GG) = 0 = 1	importance display options in the input echo in [importance] section in [region] section this function is only available when all particles are set for the same importance value
icput	(D=0) = 0 = 1	CPU time count options without count with count
ipara	(D=0) = 0 = 1	parameter display options only described parameters all parameters
nwsors	(D=0)	write down the information on nwsors source particles on file(6)

CPU time counting is not available by default, for saving calculation time. If you want to know CPU time for each process, set icput=1.

If you set ipara=1, you can confirm all parameters in the *PHITS* code.

4.2.11 Output option (4)

Table 4.12: parameter 12

parameter	value	explanation
ivoxel	(D=0)	read and write voxel data in binary
	=0	not using file(18)
	=1	read from voxel data in binary file(18)
	=2	write down voxel data in binary on file(18)
dumpall	(D=0)	dumpall option
	= 0	no dump
	= 1	write down all information on file(15) as binary data
	= -1	write down all information on file(15) as ASCII data
idpara	(D=3)	dumpall file name option in the parallel calculation
	= 0	/wk/uname/file-name of file(15)
	= 1	/wk/uname/file-name of file(15)+(PE number)
	= 3	file-name of file(15)+(PE number)
	/wk/uname/	/wk/ is the default directory name
		uname is a user-name read in from environment variable LOGNAME

For time shortening, you can use ivoxel parameter. When you perform *PHITS* calculation with ivoxel=2, voxel data are output in file(18) in binary and then the calculation is stopped (continued until ver. 2.30). From the next calculation with ivoxel=1, a process of the data output is omitted and the calculation time is shortened.

By icntl=12, *PHITS* re-calculates whole transport by reading the information from dumpall file, which is created if you use the dumpall option. The re-calculation can describe whole transport events which were calculated before. One needs the same input file as used in the previous calculation. maxcas and maxbch cannot be changed, but are read from the file. It is very powerful when you want to calculate different tallies which are not used in the previous calculation. However, please be careful that the calculation with the dumpall option may create huge dumpall file. This option is only available for GG geometry.

In the parallel computing, files corresponding to each PE (Processor Element) are created for writing and reading dumped data. If you set idpara =0 or 1, a file is made in the directory named by /wk/uname/ on each of the nodes. If you set idpara =1 or 3, the each IP number is put at the end of the filename. The each PE writes down its result on only the corresponding file, and reads it from the same file in the re-calculation.

4.2.12 About geometrical errors

Table 4.13: parameter 13

parameter	value	explanation
nlost	(D=10)	acceptable value against lost particle (per 1 PE)
igerr	(D=0)	number of recovery for region error
igchk	(D=0)	=0: no region check =1: check region setting flight mesh to deltb after region-crossing
deltb	(D=1.e-5)	flight mesh (cm) after region-crossing with igchk=1 It is also a distance from a region boundary where particle is created by the importance, and the forced collision
deltm	(D=20.12345)	A maximum of flight mesh (cm).
deltc	(D=2.012345)	A maximum of flight mesh (cm) for charged particles with nedisp=1.
delt0	(D=0.1)	A minimum of flight mesh (cm) with nspred=1 or imagnf=1. Note that when you want to describe a scattering process on thin films less than 1cm, you should set delt0 to be 1/10 of its thickness.
deltg	(D=1.0)	A maximum of flight mesh (cm) on the magnetic field.
deltt	(D=1.0)	A maximum of flight time (msec) on the time dependent magnetic field.

4.2.13 Input-output file name

Table 4.14: parameter 14

parameter	value	explanation
file(2)	(D=cgview.in)	CGVIEW input file name when icntl=2
file(3)	(D=cgview.set)	CGVIEW setup file name when icntl=2 This file becomes CGVIEW setup
file(4)	(D=marspf.in)	MARS-PF input file name when icntl=4
file(6)	(D=phits.out)	Summary output file name. If not specified, standard output
file(7)	(D=xsdir)	cross section directory file name
file(11)	(D=nuclcal.out)	nuclear reaction output file name
file(12)	(D=fort.12)	cut off neutron output file name
file(13)	(D=fort.13)	cut off γ-ray output file name
file(10)	(D=fort.10)	cut off proton output file name
file(14)	(D=trxcrd.dat)	γ decay data file name when igamma=1 path name+trxcrd.dat is required
file(15)	(D=dumpall.dat)	dump file name for dumpall=1 option
file(18)	(D=voxel.bin)	file name when you use ivoxel=1, 2
file(19)	(D=GDRxsec.inp)	Cross section data of nuclear giant resonance when ipngdr=1. Path name+GDRxsec.inp is required. GDRxsec.inp is included in the data folder of the *PHITS* package.

file(7) must be written with full pathname.

4.2.14 Others

Table 4.15: parameter 15

parameter	value	explanation
`inucr`	`(D=1)`	nuclear reaction options
	`= 1`	double differential cross section calculation
	`= 2`	total, elastic, non-elastic cross section output
	`= 3`	non-elastic cross section calculation
	`= 4`	angular distribution of elastic scattering
	`= 5`	pp, np, π-p cross section output
	`= 6`	pp, np, π-p, cross section calculation
`idam(i)`	integer	user defined integer variable
`rdam(i)`	real*8	user defined real variable
	`i = 1 - 100`	These values can be used in the *PHITS* code by `common /userp/ idam(100), rdam(100)`

Nuclear reaction calculation mode by `icntl=1` is under developing at present.

4.2.15 Physical parameters for low energy neutron

The following parameters correspond to PHYS (Energy Physics Cutoff Card) for neutron less than 20 MeV.

Table 4.16: parameter 16

parameter	value	explanation
emcnf	(D=0.0)	threshold energy for neutron capture (MeV) implicit capture is considered above this energy analog capture is considered below this energy
iunr	(D=0)	fixed 0 at present
dnb	(D=-1) =-1 = 0 > 0	number of delayed neutron by fission natural sampling no delayed neutron number of neutrons

4.2.16 Physical parameters for photon

The following parameters correspond to PHYS (Energy Physics Cutoff Card) for photon.

Table 4.17: parameter 17

parameter	value	explanation
emcpf	(D=100)	maximum energy for the detail model for photon (MeV)
ides	(D=0) = 0 = 1	electron creation options by photon create electron or brems.photon not create electron
nocoh	(D=0) = 0 = 1	coherent scattering options for photon with coherent scattering without coherent scattering

4.2.17 Physical parameters for electron

The following parameters correspond to PHYS (Energy Physics Cutoff Card) for electron.

Table 4.18: parameter 18

parameter	value	explanation
iphot	(D=0)	photon creation options by electron
	= 0	create photon
	= 1	not create photon
ibad	(D=0)	angular distribution option for brems.
	= 0	full brems. tabular angular distribution
	= 1	simple brems. angular distribution approximation
istrg	(D=0)	straggling
	= 0	sampled straggling for electron energy loss
	= 1	expected-value straggling for electron energy loss
bnum	(D=1)	brems. photon
	= 0	not create brems. photon
	> 0	number of analog brems. photons
xnum	(D=1)	x-ray photon
	= 0	not create x-ray photon
	> 0	number of analog x-ray photons
enum	(D=1)	secondary electron
	= 0	not create secondary electron
	> 0	number of analog secondary electrons
numb	(D=0)	brems. process
	= 0	nominal brems. production
	> 0	produce brems. on each substep

4.2.18 Dumpall option

By `icntl=12`, *PHITS* re-calculates whole transport by reading the information from dumpall file, which is created if you use the `dumpall` option. The re-calculation can describe whole transport events which were calculated before. One needs the same input file as used in the previous calculation. `maxcas` and `maxbch` cannot be changed, but are read from the file. It is very powerful when you want to calculate different tallies which are not used in the previous calculation. However, please be careful that the calculation with the `dumpall` option may create huge dumpall file. This option is only available for GG geometry.

The dumped data written in binary can be not used on the other computer. The data sequence and meaning are given in the following.

(1) NCOL
NCOL is an intrinsic variable in the program and denotes identification of process.

NCOL		
1	:	start of calculation
2	:	end of calculation
3	:	end of a batch
4	:	source
5	:	detection of geometry error
6	:	recovery of geometry error
7	:	termination by geometry error
8	:	termination by weight cut-off
9	:	termination by time cut-off
10	:	geometry boundary crossing
11	:	termination by energy cut-off
12	:	termination by escape or leakage
13	:	(n,x) reaction
14	:	(n,n'x) reaction
15	:	sequential transport only for tally

When NCOL=1, 2, 3, the output is finished. The followings are for NCOL≥4.

(2) NOCAS, NOBCH, RCASC, RSOUIN
These four data are written only for NCOL=4 and their meaning are,

NOCAS	:	current history number in this batch
NOBCH	:	current batch number
RCASC	:	real number of NOCAS+maxcas*(NOBCH-1)
RSOUIN	:	sum of the weight of source particle

(3) NO, MAT, ITYP, KTYP, JTYP, MTYP, RTYP, OLDWT
These mean

NO	:	cascade id in this history
MAT	:	material id
ITYP	:	particle type
KTYP	:	particle kf-code
JTYP	:	charge number of the particle
MTYP	:	baryon number of the particle
RTYP	:	rest mass of the particle (MeV)
OLDWT	:	weight of the particle at (x,y,z)

(a) QS
This data is written only for ITYP=12, 13, namely electron and positron. QS is dE/dx for electrons.

QS	:	dE/dx of electron at (x,y,z)

(4) IBLZ1, IBLZ2, ILEV1, ILEV2
These mean

IBLZ1	:	cell id at (x,y,z)
IBLZ2	:	cell id after crossing
ILEV1	:	level structure id of the cell at (x,y,z)
ILEV2	:	level structure id of the cell after crossing

(a) ILAT1
This is a variable of level structure of cell. The next data are written only for ILEV1>0 as

```
write(io) ( ( ILAT1(i,j), i=1,5 ), j=1,ILEV1 )
```

(b) ILAT2
This is a variable of level structure of cell. The next data are written only for ILEV2>0 as

```
write(io) ( ( ILAT2(i,j), i=1,5 ), j=1,ILEV2 )
```

(5) COSTH, UANG(1), UANG(2), UANG(3), NSURF
These variables mean as follows. These had come to be output from ver. 2.30.

COSTH	:	cosine of an angle of incidence in a surface crossing
UANG(1,2,3)	:	x,y,z component of a normal vector of its surface, respectively
NSURF	:	internal number of the surface Note that this is different from the surface number defined in the [surface] section

(6) NAME, NCNT(1), NCNT(2), NCNT(3)
These mean

NAME	:	collision number of the particle
NCNT(1,2,3)	:	values of counter 1, 2, and 3

(7) WT, U, V, W
These mean

WT	:	weight of the particle at (xc,yc,zc)
U, V, W	:	unit vector of momentum of the particle

(8) E, T, X, Y, Z
These mean

E	:	energy of the particle at (x,y,z) (MeV)
T	:	time of the particle at (x,y,z) (nsec)
X, Y, Z	:	position coordinate of the preceding event point (cm)

(9) EC, TC, XC, YC, ZC
These mean

EC	:	energy of the particle at (xc,yc,zc) (MeV)
TC	:	time of the particle at (xc,yc,zc) (nsec)
XC, YC, ZC	:	position coordinate of the particle (cm)

(10) SPX, SPY, SPZ
These mean

SPX, SPY, SPZ	:	unit vector of spin direction of the particle

(11) NZST
This is charge state of the particle.

(12) NCLSTS
This variable is written only for NCOL=13, 14, collision case, and means the number of produced particle and nucleus. The next data are written for NCLSTS>0 case.

(a) MATHZ, MATHN, JCOLL, KCOLL
These mean

MATHZ	:	Z number of the mother nucleus
MATHN	:	N number of the mother nucleus
JCOLL	:	reaction type id1
KCOLL	:	reaction type id2

JCOLL and KCOLL indicate the following meaning.

JCOLL

0	:	nothing happen
1	:	Hydrogen collisions
2	:	Particle Decays
3	:	Elastic collisions
4	:	High Energy Nuclear collisions
5	:	Heavy Ion reactions
6	:	Neutron reactions by data
7	:	Photon reactions by data
8	:	Electron reactions by data
9	:	Proton reactions by data
10	:	Neutron event mode
11	:	delta ray production

KCOLL

0	:	normal
1	:	high energy fission
2	:	high energy absorption
3	:	low energy n elastic
4	:	low energy n non-elastic
5	:	low energy n fission
6	:	low energy n absorption

(b) ICLUSTS, JCLUSTS, QCLUSTS, JCOUNT
These variables have a array and denote the information on the produced particle and nucleus.

```
do i = 1, NCLSTS
   write(io) ICLUSTS(i)
   write(io) ( JCLUSTS(j,i), j=0,7)
   write(io) ( QCLUSTS(j,i), j=0,12)
   write(io) ( JCOUNT(j,i),  j=1,3)
end do
```

These mean

ICLUSTS kind of particle
- 0 : nucleus
- 1 : proton
- 2 : neutron
- 3 : pion
- 4 : photon
- 5 : kaon
- 6 : muon
- 7 : others

JCLUSTS(i)
- i = 0 : angular momentum
- = 1 : proton number
- = 2 : neutron number
- = 3 : ityp
- = 4 : status of the particle 0: real, <0 : dead
- = 5 : charge number
- = 6 : baryon number
- = 7 : kf code

QCLUSTS(i)
- i = 0 : impact parameter
- = 1 : px (GeV/c)
- = 2 : py (GeV/c)
- = 3 : pz (GeV/c)
- = 4 : $e_{tot} = \sqrt{p^2 + m^2}$ (GeV)
- = 5 : rest mass (GeV)
- = 6 : excitation energy (MeV)
- = 7 : kinetic energy (MeV)
- = 8 : weight
- = 9 : time (nsec)
- = 10 : x coordinate (cm)
- = 11 : y coordinate (cm)
- = 12 : z coordinate (cm)

4.2.19 Event Generator Mode

For Event Generator mode, one should define dmax(2) appropriately, since we need the information from the data base as mentioned above. In the special statistical decay model, we use the detail information on the level structure near the ground state for particle and photon emission. For this, we need `file(14)=trxcrd.dat` and `igamma=1`. We have developed the special statistical decay model based on GEM. Then one should need to specify `nevap=3`. In this mode, the effect of thermal motion of material. It means that we always assume `T=0` in this mode. These conditions are automatically set if you specify `e-mode=1`.

By this mode, we can obtain the following new observables, which cannot be detected without this mode. First, the deposition energy distribution in [t-heat] tally is available for low energy regime below `dmax(2)`. Second, in [t-yield] and [t-product], we can tally the yield and product quantities below `dmax(2)`. Third, the heat from neutrons is usually obtained from Kerma factors in the data base region. In this mode, the heat from neutrons is zero, but the heat is calculated from energy loss of all charged particles and nucleus. Fourth, DPA values is obtained even for the energy below `dmax(2)` without DPA data base.

4.3 [S o u r c e] section

You can set source information in this section. The source type is specified by the number of "`s-type = N`".

Table 4.19: source type

source type	explanation
`s-type = 1`	cylinder (or circle, pencil)
`s-type = 4`	cylinder with energy distribution
`s-type = 2`	rectangular solid (or rectangle)
`s-type = 5`	rectangular solid with energy distribution
`s-type = 3`	Gaussian (x,y,z independent)
`s-type = 6`	Gaussian with energy distribution (x,y,z independent)
`s-type = 7`	generic parabola (x,y,z independent)
`s-type = 8`	generic parabola with energy distribution (x,y,z independent)
`s-type = 9`	sphere or spherical shell
`s-type = 10`	sphere or spherical shell with energy distribution
`s-type = 11`	uniform distribution in a phase space vertical with beam direction
`s-type = 12`	reading the data from decay-turtle output
`s-type = 13`	Gaussian (x-y plane)
`s-type = 14`	Gaussian with energy distribution (x-y plane)
`s-type = 15`	generic parabola (x-y plane)
`s-type = 16`	generic parabola with energy distribution (x-y plane)
`s-type = 17`	reading dump file
`s-type = 100`	user definition source edit the usrsors.f and compile the *PHITS*

4.3.1 <Source> : Multi-source

By this multi-source function, one can define plural sources specified by `s-type`. Each source begins <`source`> = *number*, which defines the relative weight of the multi-sources. By `totfact =` , a global normalization is defined.

Table 4.20: multi-source

parameter	explanation
`<source> =`	defines a multi-source, the relative weight is defined by this number
`totfact =`	(D=1) global normalization factor If this is given by a positive number, the source particle is generated according to this ratio. If negative, the same particle is generated in each multi-source section changing the weight according to the ratio.

4.3.2 Common parameters

Common parameters for each source type are shown below. The order of the parameters in the source section is free. If a parameter has a default value (D=***), the parameter can be omissible. The energy of d, t, α and nucleus is specified by [MeV/nucleon].

Table 4.21: common source parameters

parameter	explanation
proj =	projectile : see Table3.4 for specification
t-type = 0, 1, 2	(D=0) time distribution 0: no time-distribution, t=0.0 1: rectangle distribution 2: Gaussian distribution
t0 =	(D=0.0) center of time when t-type = 1 (ns)
tw =	FWHM of time distribution (ns)
tn =	number of time distribution
td =	interval of time distribution (ns)
tc =	(D=10×tw) cut off time when Gaussian distribution t-type=2 (ns)
sx =	(D=0) x-component of spin
sy =	(D=0) y-component of spin
sz =	(D=0) z-component of spin
reg =	(D=all) The source region can be restricted within overlaps between regions defined by each s-type and those specified by this parameter. The format is as reg = { 1 - 5 } 10 34. You can use the lattice and universe frame as reg = (6 < 10[1 0 0] < u=3). See the section about tally region specification for details.
ntmax =	(D=1000) maximum re-try number when reg is specified.
trcl =	(D=none) transform number, or definition of transform
wgt =	(D=1.0) weight of source particle
factor =	(D=1.0) normalization of source particle
izst =	(D=charge of particle specified with proj=) Charge state of source particle.

A projectile direction is specified by 3 parameters: dir, phi, and dom. The relation between these is shown in Fig. 4.1. The direction is noted by a thick arrow. The dir is a direction cosine against the *z* axis. The phi is an azimuthal angle from the *x* axis in degree. If you do not set it, a value of the azimuthal cosine is selected randomly. Using the parameter dom spreads out the direction determined by dir and phi by solid angle $2\pi(1-\cos\psi)$, where ψ = dom given in degree. In the $PHITS$ calculation, the angle is given randomly within the solid angle.

When you set dir=all, the direction of the source beam becomes isotropic. If you want to use any angular distribution, a subsection started from a-type is required, in which you should give the distribution by numerical data or analytic functions.

In s-type=9 and 10, the definition of dir is different. In s-type=11 and 12, you can set only dir= 1 or −1.

You can use the transform of coordinate by trcl parameter which specify the transform number or the definition of transformation itself. The relation of wgt and factor is reciprocal.

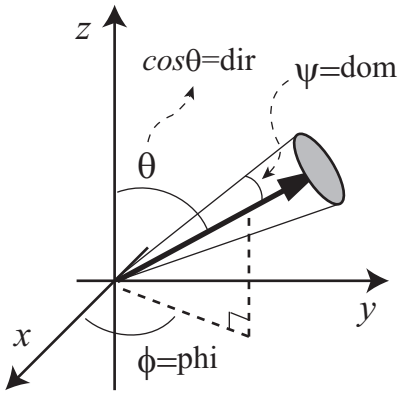


Figure 4.1: Source direction and parameters dir, phi, dom.

If the spin is not defined or zero, the neutron goes into the magnetic field without spin. In this case, the initial spin of neutron is determined at the entrance of the magnetic field by the direction of the magnetic field and the polarization factor. If the spin is defined in this section, the neutron goes into the magnetic field with the spin direction irrespective of the direction of the magnetic field nor polarization.

4.3.3 Cylinder distribution source

Parameters for cylinder source are shown below. The order of parameters is free. If a parameter has a default value (D=***), the parameter can be omissible.

Table 4.22: parameters for cylinder source

s-type = 1, 4	cylinder or circle source
x0 =	(D=0.0) x coordinate of center position of cylinder source (cm)
y0 =	(D=0.0) y coordinate of center position of cylinder source (cm)
z0 =	minimum z of cylinder source (cm)
z1 =	maximum z of cylinder source (cm), (when z1=z0, circle plane source)
r0 =	radius of cylinder source, (when r0=0.0, pencil source) (cm)
r1 =	(D=0.0) inner radius for inner void of cylinder
dir =	direction cosine of projectile against z axis
	If you set all, it is isotropic
	If you set data, a-type subsection is necessary
phi =	(D=none; random) azimuthal angle (degree)。
dom =	(D=0.0) solid angle (degree)。
	= -1 ; $\cos^2$ bias distribution
e0 =	projectile energy (s-type=1) (MeV)

4.3.4 Rectangular solid distribution source

Parameters for rectangular solid source are shown below. The order of parameters is free. If a parameter has a default value (D=***), the parameter can be omissible.

Table 4.23: parameters for rectangular solid source

s-type = 2, 5	rectangular solid and rectangle source
x0 =	minimum x coordinate (cm)
x1 =	maximum x coordinate (cm)
y0 =	minimum y coordinate (cm)
y1 =	maximum y coordinate (cm)
z0 =	minimum z coordinate (cm)
z1 =	maximum y coordinate (cm), when (z1=z0), rectangle source
dir =	direction cosine from z axis
	If you set all, isotropic
	If you set data, a-type subsection is necessary
phi =	(D=none; random) azimuthal angle (degree)。
dom =	(D=0.0) solid angle (degree)。
	= -1 ; $\cos^2$ bias distribution
e0 =	projectile energy (s-type=2) (MeV)

4.3.5 Gaussian distribution source (x,y,z independent)

This Gauss distribution is consist of independent Gaussian in each x, y, z direction. Parameters for Gaussian source are shown below. The order of parameters is free. If a parameter has a default value (D=***), the parameter can be omissible.

Table 4.24: parameters for Gaussian source

s-type = 3, 6	Gaussian source
x0 =	(D=0.0) x coordinate of Gaussian center (cm)
x1 =	FWHM in x direction (cm)
y0 =	(D=0.0) y coordinate of Gaussian center (cm)
y1 =	FWHM in y direction (cm)
z0 =	(D=0.0) z coordinate of Gaussian center (cm)
z1 =	FWHM in z direction (cm)
dir =	direction cosine from z axis
	If you set all, isotropic
	If you set data, a-type subsection is necessary
phi =	(D=none; random) azimuthal angle (degree)。
dom =	(D=0.0) solid angle (degree)。
	= -1 ; $\cos^2$ bias distribution
e0 =	projectile energy(s-type=3) (MeV)

4.3.6 Generic parabola distribution source (x,y,z independent)

This generic parabola distribution is consist of independent parabola in each x, y, z direction. Parameters for generic parabola source are shown below. The order of parameters is free. If a parameter has a default value (D=***), the parameter can be omissible.

Table 4.25: parameters for generic parabola distribution

s-type = 7, 8	generic parabola source
x0 =	(D=0.0) x coordinate of X-parabola center (cm)
x1 =	X-parabola width(cm)
y0 =	(D=0.0) y coordinate of Y-parabola center (cm)
y1 =	Y-parabola width(cm)
z0 =	minimum z of parabola (cm)
z1 =	maximum z of parabola (cm)
rn =	(D=2) order of generic parabola
dir =	direction cosine from z axis
	If you set all, isotropic
	If you set data, a-type subsection is necessary
phi =	(D=none; random) azimuthal angle (degree)。
dom =	(D=0.0) solid angle (degree)。
	= -1 ; $\cos^2$ bias distribution
e0 =	projectile energy (s-type=7) (MeV)

4.3.7 Gaussian distribution source (x-y plane)

This source is a Gaussian distribution in x-y plane. Parameters for Gaussian source are shown below. The order of parameters is free. If a parameter has a default value (D=***), the parameter can be omissible.

Table 4.26: parameters for Gaussian source

s-type = 13, 16	Gaussian source
x0 =	(D=0.0) x coordinate of Gaussian center (cm)
y0 =	(D=0.0) y coordinate of Gaussian center (cm)
r1 =	FWHM in x-y plane (cm)
z0 =	minimum z coordinate (cm)
z1 =	(D=z0) maximum z coordinate (cm)
dir =	direction cosine from z axis
	If you set all, isotropic
	If you set data, a-type subsection is necessary
phi =	(D=none; random) azimuthal angle (degree)
dom =	(D=0.0) solid angle (degree)
	= -1 ; $\cos^2$ bias distribution
e0 =	projectile energy(s-type=13) (MeV)

4.3.8 Generic parabola distribution source (x-y plane)

This source is a generic parabola distribution in x-y plane. Parameters for generic parabola source are shown below. The order of parameters is free. If a parameter has a default value (D=***), the parameter can be omissible.

Table 4.27: parameters for generic parabola distribution

s-type = 15, 16	generic parabola source
x0 =	(D=0.0) x coordinate of X-parabola center (cm)
y0 =	(D=0.0) y coordinate of Y-parabola center (cm)
r1 =	parabola width in x-y plane (cm)
z0 =	minimum z of parabola (cm)
z1 =	maximum z of parabola (cm)
rn =	(D=2) order of generic parabola
dir =	direction cosine from z axis
	If you set all, isotropic
	If you set data, a-type subsection is necessary
phi =	(D=none; random) azimuthal angle (degree)。
dom =	(D=0.0) solid angle (degree)。
	= -1 ; $\cos^2$ bias distribution
e0 =	projectile energy (s-type=15) (MeV)

4.3.9 Sphere and spherical shell distribution source

Parameters for sphere and spherical shell source are shown below. The order of parameters is free. If a parameter has a default value (D=***), the parameter can be omissible.

Table 4.28: parameters for sphere and spherical shell source

s-type = 9, 10	sphere and spherical shell source
x0 =	(D=0.0) x coordinate of sphere center (cm)
y0 =	(D=0.0) y coordinate of sphere center (cm)
z0 =	(D=0.0) z coordinate of sphere center (cm)
r1 =	inside radius (cm). If r1=0, sphere source
r2 =	outside radius (cm)
dir =	direction dir = 1.0 : outgoing from the center with normal line direction dir = -1.0 : inverse direction with dir=1.0 dir = all : isotropic dir = -all : inverse direction against dir=1.0, and with cosine distribution. This is used for volume and area calculation with $\cos^2$ bias Dir = iso : uniform distribution on a circle of radius r2 on a spherical shell of radius r1 with the direction toward the center of the sphere. In the case of r1=r2, the result is almost the same as dir = -all, but an effect of the weight on it is not included. Therefore, using the condition, you can obtain the variance of deposition energies on the [t-deposit] tally with output = deposit.
e0 =	projectile energy (s-type=9) (MeV)

When you use the source type s-type=9 for volume and area calculation, you should set as dir = -all, r1 = r2 . And, dir = iso also gives the same result.

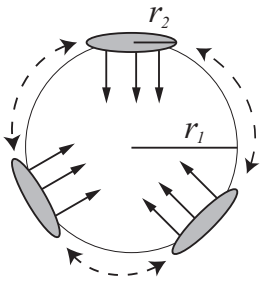


Figure 4.2: Schematic image of the source in the case of dir=iso.

4.3.10 s-type = 11

This is a uniform distribution source in a phase space which is vertical with beam direction. Parameters for this source type are shown below. The order of parameters is free. If a parameter has a default value (D=***), the parameter can be omissible.

Table 4.29: parameters for s-type = 11

s-type = 11	uniform distribution in a phase space vertical with beam direction
x0 =	(D=0.0) x coordinate of beam center (cm)
x1 =	ratio of (maximum radius)/(minimum radius) for x direction (cm/mrad)
y0 =	(D=0.0) y coordinate of beam center (cm)
y1 =	ratio of (maximum radius)/(minimum radius) for y direction (cm/mrad)
z0 =	minimum z (cm)
z1 =	maximum z (cm)
rx =	gradient of ellipse in a phase space on x direction (rad)
ry =	gradient of ellipse in a phase space on y direction (rad)
wem =	emittance (π cm $\times$ mrad)
dir =	direct cosine (1 or -1 only)
e0 =	projectile energy (MeV)

4.3.11 s-type = 12

In this source type, decay-turtle output is read as source. Parameters for this source type are shown below. The order of parameters is free. If a parameter has a default value (D=***), the parameter can be omissible.

The input file is rewinded and re-used from the first particle again, if all of source in decay-turtle is read before the calculation finishes.

Table 4.30: parameters for s-type = 12

s-type = 12	decay-turtle reading
x0 =	(D=0.0) x coordinate offset of beam (cm)
y0 =	(D=0.0) y coordinate offset of beam (cm)
z0 =	(D=0.0) z coordinate offset of beam (cm)
dir =	direction cosine(1 or -1 only)
file =	decay-turtle filename (with full pathname)

The format of `decay-turtle` is double precision, and ascii, and each record is as

```
xp, xq, yp, yq, e0, wt0, pz0
```

Table 4.31: decay-turtle data

variable	explanation
xp, yp	incoming position of beam particle (cm)
xq, yq	angle against vertical face with beam direction (mrad)
e0	momentum of beam particle (GeV/c)
wt0	weight of beam particle
pz0	polarizing of beam particle (be not in use)

4.3.12 Reading dump file

In this source type, the data of the dump file is read in as source. Parameters for the `type = 17` are shown below. The order of parameters is free. If a parameter has a default value (D=***), the parameter can be omissible.

The dump file is rewinded and re-used from the first data again, if all of source in the dump file is read before the calculation finishes.

Table 4.32: parameters for dump file source

`s-type = 17`	reading dump file
`file =`	dump filename (with full pathname)
`dump =`	number of dump data. if it is negative, data is written by Ascii.
(next line)	identification of dump data
(omissible)	If below parameters are specified, these values have priority over the dump data. If the dump data does not include the following data, one should specify the parameters.
`x0 =`	minimum x coordinate (cm)
`x1 =`	maximum x coordinate (cm)
`y0 =`	minimum y coordinate (cm)
`y1 =`	maximum y coordinate (cm)
`z0 =`	minimum z coordinate (cm)
`z1 =`	maximum z coordinate (cm)
`sx =`	(D=0) x-component of spin
`sy =`	(D=0) y-component of spin
`sz =`	(D=0) z-component of spin
`dir =`	direction cosine from z axis
	If you set `all`, isotropic
	If you set `data`, `a-type` subsection is necessary
`phi =`	(D=none; random) azimuthal angle (degree)
`dom =`	(D=0.0) solid angle (degree)
	= -1 ; $\cos^2$ bias distribution
`e0 =`	projectile energy (MeV)
`e-type =`	type of energy distribution
`wgt =`	(D=1.0) weight of source particle
`factor =`	(D=1.0) normalization of source particle
(omissible)	one can also specify below parameters
`t-type =`	(D=0) time distribution
`reg =`	(D=all) specify the region
`ntmax =`	(D=1000) maximum re-try number when `reg` is specified
`trcl =`	(D=none) transform number, or definition of transform

By the parameter of "`dump =`", the number of the dump data in one record is specified. If this number is given by positive number, the data is read as binary data. If negative, the data is read as asci data. In next line, the data sequence of one record is described. The relation between the physical quantities and id number is the followings,

Table 4.33: id number of dump data (1)

physical quantities	kf	x	y	z	u	v	w	e	wt	time	c1	c2	c3	sx	sy	sz
id number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Table 4.34: id number of dump data (2)

physical quantities	name	nocas	nobch	no
id number	17	18	19	20

Here kf means the kf-code of the particles (see Table 3.4), x, y, z are coordinates (cm), u, v, w denote the unit vectors of the direction of the particle, e is the energy (MeV, or MeV/nucleon for nucleus), wt is the weight, time is the initial time (ns), c1, c2, c3 are the values of counters, and sx, sy, sz are the unit vectors of the direction of spin, respectively. name is a collision number of the particle, nocas is a current history number of this batch, nobch is a current batch number, no is a cascade id in this history. These are assumed as real*8 for the binary data, *n*(1p1e24.15) data format for the ascii data.

For an example, one record has 9 data as

```
kf  e  wt  x  y  z  u  v  w
```

To read this data, we write the parameters as

```
dump = 9
  1  8  9  2  3  4  5  6  7
```

4.3.13 User definition source

If you edit usrsors.f, you can use your original source function by s-type=100. If the following parameters are set, these values have the priority. If a parameter has a default value (D=***), the parameter can be omissible.

Table 4.35: the parameters can be specified in s-type=100

s-type = 100	user definition source
	If below parameters are specified, these values have priority over the user defined data.
x0 =	minimum x coordinate (cm)
x1 =	maximum x coordinate (cm)
y0 =	minimum y coordinate (cm)
y1 =	maximum y coordinate (cm)
z0 =	minimum z coordinate (cm)
z1 =	maximum z coordinate (cm)
sx =	(D=0) x-component of spin
sy =	(D=0) y-component of spin
sz =	(D=0) z-component of spin
dir =	direction cosine from z axis
	If you set all, isotropic
	If you set data, a-type subsection is necessary
phi =	(D=none; random) azimuthal angle (degree)。
dom =	(D=0.0) solid angle (degree)。
	= -1 ; $\cos^2$ bias distribution
e0 =	projectile energy (s-type=7) (MeV)
e-type =	type of energy distribution
wgt =	(D=1.0) weight of source particle
factor =	(D=1.0) normalization of source particle
t-type =	(D=0) time distribution
reg =	(D=all) specify the region
ntmax =	(D=1000) maximum re-try number when reg is specified.
trcl =	(D=none) transform number, or definition of transform

We show a sample program of usrsors.f as following. In the first comment part, there is a list of the variables which is necessary to define the source. Next there is a list of kf-code which specifies the source particle. In the last part of the comment, the random number functions, one is an uniform random number, the other is a Gaussian random number, are shown. The first part of the program is an example of the initialization, which describes the open and close the data file. The remaining part shows a list of the variables which user should define in this subroutine.

List 4.1 ● usrsors.f

```
      subroutine usrsors(x,y,z,u,v,w,e,wt,time,name,kf,nc1,nc2,nc3,
     &                   sx,sy,sz)
*        sample subroutine for user defined source.                    *
*        variables :                                                   *
*           x, y, z : position of the source.                          *
*           u, v, w : unit vector of the particle direction.           *
*           e       : kinetic energy of particle (MeV).                *
*           wt      : weight of particle.                              *
*           time    : initial time of particle. (ns)                   *
*           name    : usually = 1, for Coulmb spread.                  *
*           kf      : kf code of the particle.                         *
*           nc1     : initial value of counter 1                       *
*           nc2     : initial value of counter 2                       *
*           nc3     : initial value of counter 3                       *
*           sx,sy,sz : spin components                                 *
*-----------------------------------------------------------------------*
*        kf code table                                                 *
*           kf-code: ityp :  description                               *
*             2212 :   1  :  proton                                    *
*             2112 :   2  :  neutron                                   *
*              211 :   3  :  pion (+)                                  *
*              111 :   4  :  pion (0)                                  *
*             -211 :   5  :  pion (-)                                  *
*              -13 :   6  :  muon (+)                                  *
*               13 :   7  :  muon (-)                                  *
*              321 :   8  :  kaon (+)                                  *
*              311 :   9  :  kaon (0)                                  *
*             -321 :  10  :  kaon (-)                                  *
*           kf-code of the other transport particles                   *
*               12 :         nu_e                                      *
*               14 :         nu_mu                                     *
*              221 :         eta                                       *
*              331 :         eta'                                      *
*             -311 :         k0bar                                     *
*            -2112 :         nbar                                      *
*            -2212 :         pbar                                      *
*             3122 :         Lanbda0                                   *
*             3222 :         Sigma+                                    *
*             3212 :         Sigma0                                    *
*             3112 :         Sigma-                                    *
*             3322 :         Xi0                                       *
*             3312 :         Xi-                                       *
*             3334 :         Omega-                                    *
*-----------------------------------------------------------------------*
*        available function for random number                          *
*           unirn(dummy) : uniform random number from 0 to 1           *
*           gaurn(dummy) : gaussian random number                      *
*                          for exp( - x**2 / 2 / sig**2 ) : sig = 1.0  *
************************************************************************
      implicit real*8 (a-h,o-z)
*-----------------------------------------------------------------------
      parameter ( pi  = 3.141592653589793d0 )
```

```
54:         data ifirst / 0 /
55:         save ifirst
56:         character filenm*50
57:   *--------------------------------------------------------------------
58:   *        example of initialization
59:   *--------------------------------------------------------------------
60:           if( ifirst .eq. 0 ) then
61:   c             filenm = 'input.dat'
62:   c             inquire( file = filenm, exist = exex )
63:   c             if( exex .eqv. .false. ) then
64:   c                write(*,*) 'file does not exist =>  ', filenm
65:   c                call parastop( 887 )
66:   c                end if
67:   c             open(71, file = file(i), status = 'old' )
68:
69:   c             close(71)
70:                 ifirst = 1
71:           end if
72:   *--------------------------------------------------------------------
73:   *        example for 3 GeV proton with z-direction
74:   *--------------------------------------------------------------------
75:              x = 0.0
76:              y = 0.0
77:              z = 0.0
78:              u = 0.0
79:              v = 0.0
80:              w = 1.0
81:              e = 3000.0
82:             wt = 1.0
83:           time = 0.0
84:           name = 1
85:             kf = 2212
86:            nc1 = 0
87:            nc2 = 0
88:            nc3 = 0
89:             sx = 0.d0
90:             sy = 0.d0
91:             sz = 0.d0
92:   *--------------------------------------------------------------------
93:         return
94:         end
```

4.3.14 Definition for energy distribution

In the energy distributed source type `s-type = 4`, `5`, `6`, `8`, `10`, `14`, and `16`, or `s-type = 17, 100` with e-type parameter, energy distribution parameters are required as shown below. For *d*, *t*, α and nucleus, this energy is expressed in units of [MeV/nucleon]. Note that the source intensity in each energy bin should be given in the energy integrated value, not the energy differential value expressed in (particle/MeV). If a parameter has a default value (D=***), the parameter can be omissible.

Table 4.36: parameters for source energy distribution (1)

parameter	explanation
`e-type = 1, (11)`	You can specify any energy distribution by giving data set of energy bins `e(i)` and probabilities of the particle generation `w(i)` by hand. The number of the particle generation in the bin is proportional to `w(i)`, and the specified energy distribution is statistically described. For 11 case, energy is given by wave length (Å).
`ne =`	number of energy group If it is given by positive number, linear interpolation is assumed in a bin. If negative, logarithmic interpolation is assumed in a bin. Data must be given from the next line by the format as `(e(i),w(i),i=1,ne), e(ne+1)`. The number of the particle generation in the each energy bin is proportional to `w(i)`. Note that the source intensity in each energy bin should be the energy integrated value, not the energy differential value expressed in (particle/MeV).
`e-type = 4, (14)`	You can specify any energy distribution by giving data set of energy bins `e(i)` and weights of the source particle w(i) by hand. The number of the particle generation is the same in all of the bins, but the weight of the particle is given proportionally with `w(i)`. Setting `w(i)` the specified energy distribution is described. You can also change the number of the generation using the generation option `p(i)`. For 14 case, energy is given by wave length (Å).
`ne =`	number of energy group data must be given from the next line by the format below If it is given by positive number, linear interpolation is assumed in a bin. If negative, logarithmic interpolation is assumed in a bin. In default (p-type=0), equal number of particle is generated in each cell. `(e(i),w(i),i=1,ne), e(ne+1)` The number of the particle generation in the each energy bin is proportional to `p(i)`.
`p-type = 0, 1`	(D=0) generation option for 0, `p(i)`=1 for all `i` is assumed without the following data for 1, p(i) must be given from the next line by the format as `(p(i),i=1,ne)`

For neutron optics, we prepare an alternative option to specify the energy as wave length. If you specify e-type=11, 12, 14, you can use wave length (Å) as an energy unit. For the other case, you can use the mathematical expressions as e0=8.180425e-8/13**2, which gives the energy of neutron with 13Åwave length.

Table 4.37: parameters for source energy distribution (2)

parameter	explanation
e-type = 2, (12)	Gaussian distribution for 12 case, energy is given by wave length (Å)
eg0 = eg1 = eg2 = eg3 =	center of Gaussian distribution (MeV) FWHM of Gaussian distribution (MeV) minimum cut off for Gaussian distribution (MeV) maximum cut off for Gaussian distribution (MeV)
e-type = 3	Maxwellian distribution : $f(x) = x^{1.5}\exp(-x/T)$
nm =	(D=-200) number of energy group If it is given by positive number, linear interpolation is assumed in a bin. If negative, logarithmic interpolation is assumed in a bin. Number of particle generation in a cell is proportional to f(x).
et0 = et1 = et2 =	temperature parameter T (MeV) minimum cut off for Maxwellian distribution (MeV) maximum cut off for Maxwellian distribution (MeV)
e-type = 7	Maxwellian distribution : $f(x) = x^{1.5}\exp(-x/T)$
nm =	(D=-200) number of energy group If it is given by positive number, linear interpolation is assumed in a bin. If negative, logarithmic interpolation is assumed in a bin. In default (p-type=0), equal number of particle is generated in each cell.
et0 = et1 = et2 =	temperature parameter T (MeV) minimum cut off for Maxwellian distribution (MeV) maximum cut off for Maxwellian distribution (MeV)
p-type = 0, 1	(D=0) generation option for 0, p(i)=1 for all i is assumed without the following data for 1, p(i) must be given from the next line by the format as (p(i),i=1,ne)
e-type = 5, (15)	energy distribution is given by f(x) for 15 case, energy is given by wave length (Å)
f(x) =	any analytical function of x, FORTRAN style one can use intrinsic functions and constants C.
nm =	number of energy group If it is given by positive number, linear interpolation is assumed in a bin. If negative, logarithmic interpolation is assumed in a bin. Number of particle generation in a cell is proportional to f(x)
eg1 = eg2 =	minimum cut off for energy distribution (MeV) maximum cut off for energy distribution (MeV)
e-type = 6, (16)	energy distribution is given by f(x) for 16 case, energy is given by wave length (Å)
f(x) =	any analytical function of x, FORTRAN style one can use intrinsic functions and constants C.
nm =	number of energy group If it is given by positive number, linear interpolation is assumed in a bin. If negative, logarithmic interpolation is assumed in a bin. In default (p-type=0), equal number of particle is generated in each cell.
eg1 = eg2 =	minimum cut off for energy distribution (MeV) maximum cut off for energy distribution (MeV)
p-type = 0, 1	(D=0) generation option for 0, p(i)=1 for all i is assumed without the following data for 1, p(i) must be given from the next line by the format as (p(i),i=1,nm)

4.3.15 Definition for angular distribution

If you set `dir = data`, angular distribution parameters are required as shown below. If a parameter has a default value (D=***), the parameter can be omissible.

Table 4.38: parameters for source angular distribution (1)

parameter	explanation
`a-type = 1, (11)`	give angle and weight by hand for 1 case, angle is given by cosine, for 11 case, given by degree number of particle generation in a cell is proportional to `w(i)`
`na =`	number of angular group If it is given by positive number, linear interpolation is assumed in a bin. If negative, logarithmic interpolation is assumed in a bin. Data must be given from the next line by the format as `(a(i),w(i),i=1,na), a(na+1)`.
`a-type = 4, (14)`	give angular and weight by hand for 4 case, angle is given by cosine, for 14 case, given by degree number of particle generation in a cell is proportional to `p(i)`
`na =`	number of angular group data must be given from the next line by the format below If it is given by positive number, linear interpolation is assumed in a bin. If negative, logarithmic interpolation is assumed in a bin. In default (q-type=0), equal number of particle is generated in each cell. `(a(i),w(i),i=1,na), a(na+1)`
`q-type = 0, 1`	(D=0) generation option for 0, `q(i)=1` for all `i` is assumed without the following data for 1, q(i) must be given from the next line by the format as `(q(i),i=1,na)`

Table 4.39: parameters for source angular distribution (2)

parameter	explanation
a-type = 5, (15) g(x) =	angular distribution is given by g(x) for 5 case, angle is given by cosine, for 15 case, given by degree any analytical function of x, FORTRAN style one can use intrinsic functions and constants C.
nn =	number of angular group If it is given by positive number, linear interpolation is assumed in a bin. If negative, logarithmic interpolation is assumed in a bin. Number of particle generation in a cell is proportional to g(x).
ag1 = ag2 =	minimum cut off for angular distribution maximum cut off for angular distribution
a-type = 6, (16) g(x) =	angular distribution is given by g(x) for 6 case, angle is given by cosine, for 16 case, given by degree any analytical function of x, FORTRAN style one can use intrinsic functions and constants C.
nn =	number of angular group If it is given by positive number, linear interpolation is assumed in a bin. If negative, logarithmic interpolation is assumed in a bin. In default (q-type=0), equal number of particle is generated in each cell.
ag1 = ag2 =	minimum cut off for angular distribution maximum cut off for angular distribution
q-type = 0, 1	(D=0) generation option for 0, q(i)=1 for all i is assumed without the following data for 1, q(i) must be given from the next line by the format as (q(i),i=1,nn)

4.3.16 Example of multi-source

We introduce an example of multi-source, which includes energy distribution and angular distribution described by analytic functions. The list of third multi-source is shown below.

List 4.2 ● Example of multi-source

```
[ S o u r c e ]
 totfact = 3
<source> = 9.72
  s-type = 4
    proj = proton
      z0 = 2
      z1 = 29
      r0 = 5
      r1 = 4
     dir = 0.0
  e-type = 6
     eg1 = 1.e-6
     eg2 = 1.e-3
     nm  = -200
    set: c10[1.e-4]
    f(x) = x**(1.5)*exp(-x/c10)
<source> = 1
  s-type = 4
    proj = photon
      z0 = 1
      z1 = 2
      r0 = 5
     dir = -1
  e-type = 5
     eg1 = 1.e-3
     eg2 = 5.e-1
     nm  = 200
     set: c10[1.e-1]
     set: c20[1.e-1/2.35482]
    f(x) = exp(-(x-c10)**2/2/c20**2)
<source> = 1
  s-type = 4
    proj = neutron
      z0 = 29
      z1 = 30
      r0 = 5
  e-type = 6
     eg1 = 1.e-2
     eg2 = 1.e+3
     nm  = -200
     set: c10[92.469]
     set: c20[5.644e+10]
    f(x) = c10/c20*exp(-sqrt(x*(x+1876))/c10)*(x+938)/sqrt(x*(x+1876))
     dir = data
  a-type = 5
     ag1 = 0
     ag2 = 1
     nn  = 200
    g(x) = exp(-(x-1)**2/0.3**2)
```

In this example, there are three source subsections started from <source>. In the first source subsection, we define a cylinder source from z=2cm to z=29cm with 5cm radius, and we set r1=4. This r1=4 means that the region inside the cylinder with radius 4cm is not included. In the next source, it is also a cylinder source from z=1cm to z=2cm with 5cm radius without r1. This is a normal thin cylinder. The last one is also a thin cylinder from z=29cm to z=30cm with 5cm radius. The numbers defined after each <source> denote the relative weight of the multi-source. In this example, the relative weight is determined by the relative volume ratio of each source. This means that the source particles are generated uniformly in each source volume. The coordinate distribution of the generated source particles is shown in Fig. 4.3 using [t-product] tally with output=source, and icntl=6.

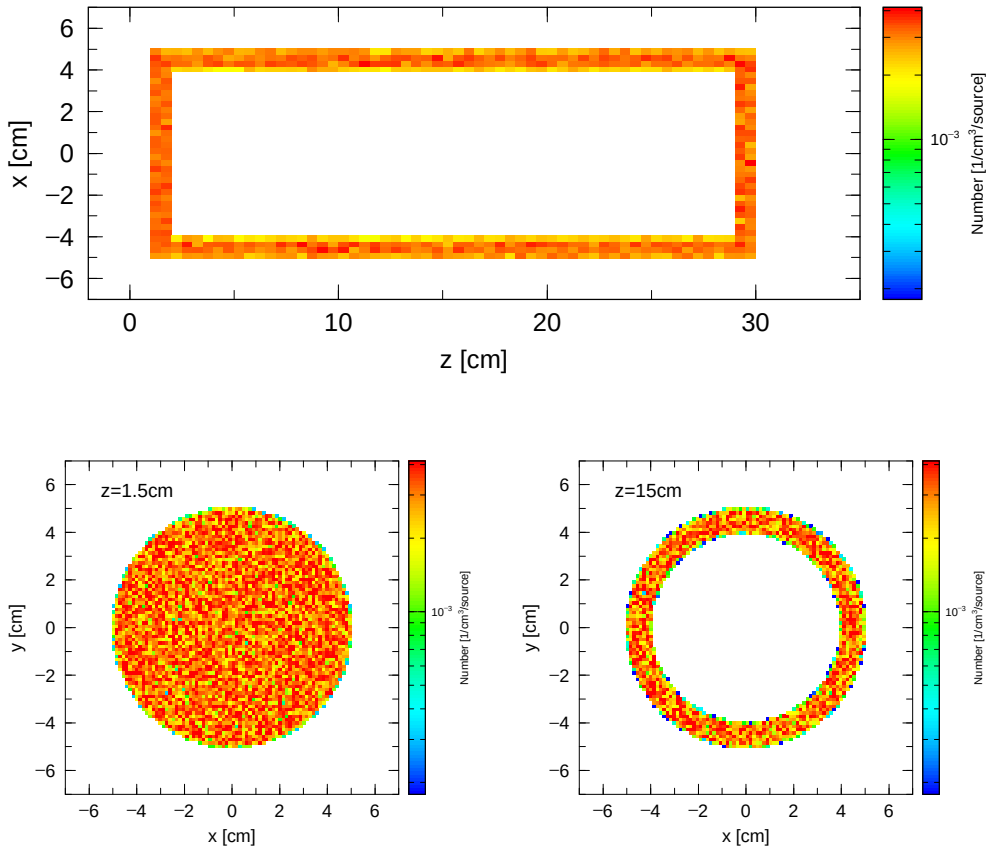


Figure 4.3: Multi-source, coordinate distribution

The source particles of the multi-source are proton, photon and neutron. In each subsection, the energy distribution of the source particle is defined as Maxwellian, Gaussian, and user defined analytical function by using the expression of those function with FORTRAN style. The first Maxwellian distribution is just equivalent to the expression by e-type=7 as

```
e-type = 7
   et0 = 1.e-4
   et1 = 1.e-6
   et2 = 1.e-3
```

The second Gaussian distribution is also equivalent to the expression by e-type=2 as

```
e-type = 2
   eg0 = 1.e-1
   eg1 = 1.e-1
   eg2 = 1.e-4
   eg3 = 5.e-1
```

These energy distributions are shown below by using [t-product] tally with output=source, and icntl=6. The result of each particle is shown in Fig. 4.4 with different colors.

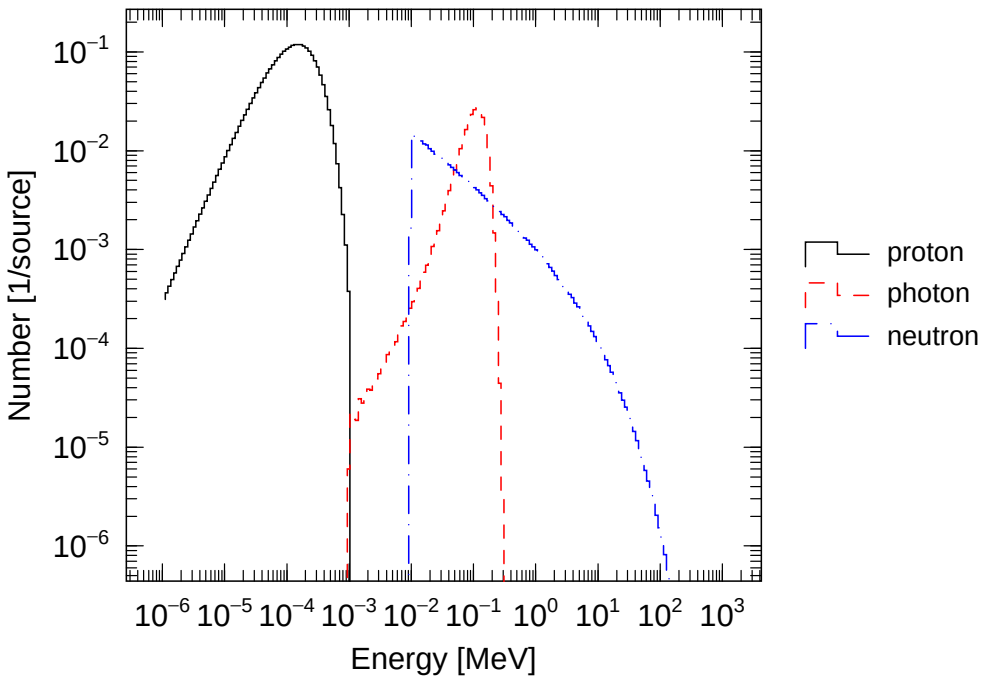


Figure 4.4: Multi-source, energy distribution

The first source has an angular distribution defined by dir=0, which means 90 degrees direction with respect to z-axis, the second one has dir=-1, 180 degrees direction, and the third one has an angular distribution defined by a-type description in which we used an analytic function for an angular distribution. The angular distribution of the third one is shown in Fig. 4.5 by using [t-cross] tally.

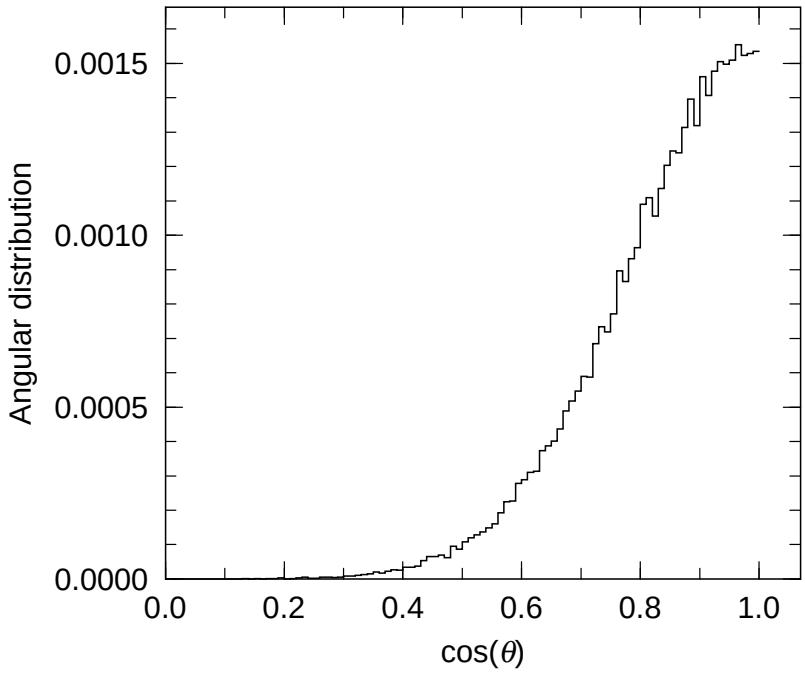


Figure 4.5: Multi-source, angular distribution

4.3.17 Duct source option

For the simulation of neutrons through the long beam-line from the moderator of spallation neutron source or the reactor to the detector room, we have prepared the following duct source options to reduce the variance of the calculations. The beam current transported through the beam-line decreases proportional to the inverse square of the distance from the moderator. This means that the current crossing the wall of the beam-line, which is called as ”wall current”, at 100 m point is six order of magnitude smaller than that at 1 m point from the source if we assume isotropic distribution of the source direction. To reduce this variance, we have introduced a special options of the source function in which the wall current of the simulation particles is equalized at any point of the beam-line by changing the importance weight of the particles to simulate a real situation of the current inside the beam-line.

We set the duct source options for `s-type = 1, 4, 2, 5`, circle and rectangle source, by `dom = -10`. The parameters for the duct source options are summarized in Table 4.40.

Table 4.40: parameters for duct source options

parameter	explanation
`dom = -10`	specify the duct source
`dl0 =`	starting z position of the beam-line from z0 (cm)
`dl1 =`	starting z position of the duct source from z0 (cm)
`dl2 =`	ending z position of the duct source from z0 (cm)
`dpf =`	portion of pass through particles at dl2
`drd =`	radius of circle beam line for s-type = 1, 4 (cm)
`dxw =`	x size of rectangle beam line for s-type = 2, 5 (cm)
`dyw =`	y size of rectangle beam line for s-type = 2, 5 (cm)

We assume circle or rectangle beam-line for `s-type = 1, 4` or `s-type = 2, 5`, respectively. `z1 = z0` and `dir = 1` are also assumed, the latter means the direction of the beam-line. If you want to change the direction of the beam-line, you should use the transformation `trcl =` *number of transformation*. The source particles are generated within the circle or rectangle region at `z0` defined by `r0` or `x0, x1, y0, y1`, for `s-type = 1, 4` or `s-type = 2, 5`, respectively. The direction of the particle is determined by the wall position where it reaches within `dl1` and `dl2` so as to equalize the wall current at any point within this region changing the importance of the particle. Overall normalization factor is defined as a number of the source particles which pass the entrance of beam-line at `dl0` originated within the same region at the source position `z0` as that at `dl0`. We normally set the number to be unit for one history if all duct wall position from `dl0` to `dl2` can see the source region at `z0`. If the source region at `z0` is larger than the area of the beam-line at `dl0`, the source particle from the outer region at `z0` is not counted as the normalization number at `dl0`. This means that the extra region at `z0` increases the current in the beam-line without changing the normalization factor. In the above argument, we assume isotropic angular distribution of the source particles within the small solid angle which covers the whole beam-lin.

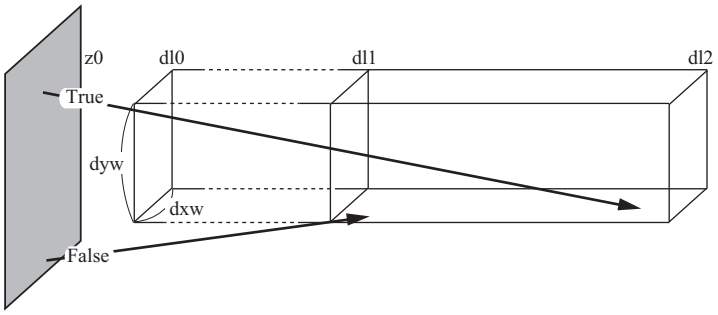


Figure 4.6: Schematic image of the duct source.

We show some example of the duct source option in the following. In the first example, we use the rectangle source and beam-line, the same size of the source and beam-line dimensions. Here we show the input for the duct source option,

List 4.3 ● duct source option, example 1

```
 1    [ S o u r c e ]
 2:
 3:   set: c1[200]    $dl0
 4:   set: c2[500]    $dl1
 5:   set: c3[5000]   $dl2
 6:   set: c4[5.0]    $x*2 at z0
 7:   set: c5[5.0]    $y*2 at z0
 8:   set: c10[5.0]   $dxw
 9:   set: c20[5.0]   $dyw
10:   set: c30[0.001] $dpf
11:
12:   s-type = 2
13:   proj   =  neutron
14:   e0     =  20.0
15:   x0     = -c4/2
16:   x1     =  c4/2
17:   y0     = -c5/2
18:   y1     =  c5/2
19:   z0     =  0.0
20:   z1     =  0.0
21:   dir    =  1.0
22:   phi    =  0.0
23:
24:   dom = -10
25:   dl0 = c1
26:   dl1 = c2
27:   dl2 = c3
28:   dxw = c10
29:   dyw = c20
30:   dpf = c30
```

In the first part of above source section, we define some constants which are necessary for the duct source option, `dl0, dl1, dl2`, size of source, `dxw, dyw, dpf`. In the second part, we define the position and xy region of the source, direction of the beam-line and the energy of source particle. In the third part, we define the duct source options. We calculated particle transport in the beam-line from 5 m up to 50 m by this duct source and the current, wall current by using the cross tally. The results are shown in Fig. 4.7 compared with an ideal case in which the current and the wall current are proportional to $1/L^2$ and $1/L^3$, respectively. The cross marker in the figure indicates the position of `dl0` and show that the current at this point is unit. The results of the duct source option agree very well with the analytical results.

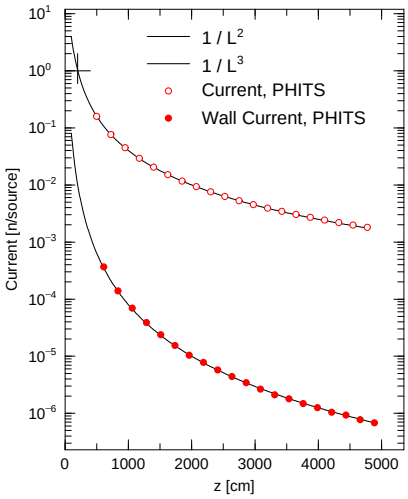


Figure 4.7: duct source option, example 1

In the next example, we changed only the size of the source from the previous example.

List 4.4 ● duct source option, example 2

```
 1    [ S o u r c e ]
 2:
 3:   set: c1[200]    $dl0
 4:   set: c2[500]    $dl1
 5:   set: c3[5000]   $dl2
 6:   set: c4[10.0]   $x*2 at z0
 7:   set: c5[10.0]   $y*2 at z0
 8:   set: c10[5.0]   $dxw
 9:   set: c20[5.0]   $dyw
10:   set: c30[0.001] $dpf
11:
12:   s-type = 2
13:   proj   =  neutron
14:   e0     =  20.0
15:   x0     = -c4/2
16:   x1     =  c4/2
17:   y0     = -c5/2
18:   y1     =  c5/2
19:   z0     =  0.0
20:   z1     =  0.0
21:   dir    =  1.0
22:   phi    =  0.0
23:
24:   dom = -10
25:   dl0 = c1
26:   dl1 = c2
27:   dl2 = c3
28:   dxw = c10
29:   dyw = c20
30:   dpf = c30
```

Figure 4.8 shows how the extra region of the source increases the current and the wall current. By this function, you can automatically treat the margin area of the moderator to the size of the cross-section of beam-line.

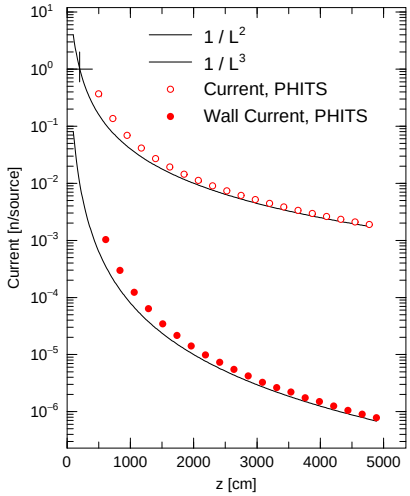


Figure 4.8: duct source option, example 2

4.4 [M a t e r i a l] section

4.4.1 Formats

Material is defined in this section. There are two formats for material definition as shown below. First, the conventional format as

```
[ M a t e r i a l ]
   MAT[ m ]
        nucleus   density
        nucleus   density
        nucleus   density
        nucleus   density
   MAT[ m ]
        keyword=value  .....
        nucleus   density
        nucleus   density
        MTm   S(a,b) identifier ....
   MAT[ m ]
          ......    .......
          ......    .......
```

In addition, other definition format can be used as

```
[ M a t e r i a l ]
   Mm   nucleus   density      nucleus   density
        nucleus   density      nucleus   density
   Mm
        keyword=value  .....
        nucleus   density      nucleus   density
        nucleus   density      nucleus   density
        MTm   S(a,b) identifier ....
   Mm
         ......    .......
         ......    .......
```

Here *m* can be specified up to material number 9999 unless over-defined.

4.4.2 Nuclide definition

The *nucleus* can be defined by various format as `208Pb, Pb-208, 82208`. Hydrogen is defined as `1H, H-1, 1001`. You can use natural isotope ratio if you use no-mass style as `Pb, 82000`.

If you want to use nuclear data, add library number (double-digit) and data class (a character a-z) after nuclide definition after period as `208Pb.24c, Pb-208.24c, 82208.24c`. In the case that you do not add such extension, *PHITS* searches the cross section directory file `file(7) (D=xsdir)` from the top line for the library number id corresponding to the nuclide, and use its data library. You can specify library number id on each material number, not on each nuclide, by `NLIB, PLIB, ELIB, HLIB` parameters in Table 4.41. Note that the specification with the above extension have higher priority than that with these parameters. Information on used data library in a *PHITS* calculation is written in the summary output file `file(6) (D=phits.out)` when `kmout=1` is set in `[parameters]` section .

If you want to use “carbon”, you should use `6000`, `6012` or `12C`. Because “c” doubles as a comment mark.

4.4.3 Density definition

Two units are available for *density* definition. if *density* takes positive value, it means particle density as [10^{24} atoms/cm^3] else if negative value, mass density [g/cm^3]. These densities can be re-defined in the [region] and [cell] sections. So if you have density definition in the [region] of [cell] section, you don't need to set density, alternatively you can set composition ratio here.

4.4.4 Material parameters

For the region in which you use nuclear data, you can set material parameters for each material as the style *keyword=value*. Anywhere you can write this parameters in the material subsection. Parameters are shown below.

Table 4.41: material parameter

parameter	value	explanation
GAS	D = 0	density effect correction to electron stopping power
	= 0	appropriate for material in the condensed (solid or liquid) state used
	= 1	appropriate for material in the gaseous state used
ESTEP	= *n*	make sub step number *n* for electron transport it is ignored when *n* is smaller than built-in default value
NLIB	= *id*	change default neutron library number *id*
PLIB	= *id*	change default photon library number *id*
ELIB	= *id*	change default electron library number *id*
HLIB	= *id*	change default proton library number *id*
COND		conductor settings
	< 0	non-conductor
	= 0	(Default) non-conductor if there exist at least 1 non-conductor, otherwise conductor
	> 0	conductor if there exist at least 1 conductor

4.4.5 $S(\alpha,\beta)$ settings

In the transport of low-energy neutrons, $S(\alpha,\beta)$ library may be required. This library can be set as follows:

```
MTm materialID
```

where m is the material number, and materialID is ID number written in "xsdir" such as lwtr.20t. See \XS\tsl\tsl-table for detailed information for these data.

4.4.6 Examples

Some examples are shown below.

List 4.5 ● material example (1)

```
 1:  [ M a t e r i a l ]
 2:  MAT[ 1 ]
 3:     1H        1.0000000E-04
 4:   208Pb       1.7238000E-02
 5:   204Pb       4.6801000E-04
 6:   206Pb       7.9430000E-03
 7:   207Pb       7.2838000E-03
 8:  MAT[ 2 ]
 9:     1H        1.0000000E-09
10:    14N        4.6801000E-05
11:    16O        7.9430000E-06
```

By default, the order is nuclide, then density. You can specify them in reverse by putting the "den" and "nuc" as,

List 4.6 ● material example (2)

```
 1:  [ M a t e r i a l ]
 2:     den            nuc      <------
 3:  MAT[ 1 ]
 4:  1.0000000E-04      1H
 5:  1.7238000E-02    208Pb
 6:  4.6801000E-04    204Pb
 7:  7.9430000E-03    206Pb
 8:  7.2838000E-03    207Pb
 9:  MAT[ 2 ]
10:  1.0000000E-09      1H
11:  4.6801000E-05     14N
12:  7.9430000E-06     16O
```

List 4.7 ● material example (3)

```
 1:  [ M a t e r i a l ]
 2:  m1     80196.49c  5.9595d-5
 3:         80198.49c  3.9611d-3
 4:         80199.49c  6.7025d-3
 5:         80200.49c  9.1776d-3
 6:         80201.49c  5.2364d-3
 7:         80202.49c  1.1863d-2
 8:         80204.49c  2.2795d-3
 9:  c      ...Be...
10:  m3     4009.37c  1.2362E-1
11:  mt3    be.01
12:  c      ...h2o (25C)...
13:  m4     1001.37c  6.6658d-2      8016.37c  3.3329d-2
14:  mt4    lwtr.01
15:  c      ...b4c (natural boron; 25%-density)...
16:  m5     6012.37c  6.8118d-3
17:         5011.37c  2.1825d-2
18:  c      ...liquid-h2 (20K)...
19:  m6     1001.49c  3.1371d-2      1011.49c  1.0457d-2
20:  mt6    orthoh.00   parah.00
```

4.5 [B o d y] section

4.5.1 formats

You can set body definition by the CG geometry in this section. In this section definition, you can write any comments after [body] section name as

[body] HIMAC_experiment.

But you can not use comments which begin off here, otherwise, the [body] section is skipped.

Parameters idbg,ibod and naz can be set at beginning of this section. If omitted, default values are used instead.

Table 4.42: body control parameters

name	value	explanation
idbg	(D=0)	debugging options
	= 0	none
	= 1	CG input echo
	= 2	CG debugging
ibod	(D=1)	body number
	= 0	omitting body number
	> 0	body number
naz	(D=0)	array size for keeping data for tangent region By default, it is set already at least for 5 regions. If you need more than 5, give number (generally, input 0)
	= 0	general value
	> 0	maximum number of tangent region

These parameters can be written in a line as

```
idbg = 0 ;  ibod = 1 ;  naz =  0
```

Body should be defined by body name symbol (sym), body number (num), and body shape definition (def). These order can be changed, but body shape definition (def) must be put at the last of line. Default order is

```
num   sym   def
```

If you have additional column you can skip the column by the non declaration. Details for body symbol and body shape definition is quite same as the CG geometry system, see manuals about the CG. If you skip to specify body number, PHITS gives body number in order from top automatically. In the case body shape definition can not be written in 1 line, you can write in the next line without no symbols. Some examples are shown below.

4.5.2 Examples

List 4.8 ● body example (1)

```
1:  [ B o d y ]   body example 1
2:       1   rpp    -7.5000000E+00  7.5000000E+00
3:                  -7.5000000E+00  7.5000000E+00
4:                  -1.0000000E+01  1.0000000E+01
5:       2   sph     0.0000000E+00  0.0000000E+00  0.0000000E+00  9.9900000E+01
6:       3   sph     0.0000000E+00  0.0000000E+00  0.0000000E+00  1.0000000E+02
```

List 4.9 ● body example (2)

```
1:  [ B o d y ]   body example 2
2:   idbg = 0 ;  ibod = 0 ;  naz =  0
3:     rpp    -7.5000000E+00  7.5000000E+00
4:            -7.5000000E+00  7.5000000E+00
5:            -1.0000000E+01  1.0000000E+01
6:     sph     0.0000000E+00  0.0000000E+00  0.0000000E+00  9.9900000E+01
7:     sph     0.0000000E+00  0.0000000E+00  0.0000000E+00  1.0000000E+02
```

List 4.10 ● body example (3)

```
1:  [ B o d y ]   body example 3
2:   idbg = 0 ;  ibod = 1 ;  naz =  0
3:    num   sym     def
4:      1   rpp    -7.5000000E+00  7.5000000E+00
5:                 -7.5000000E+00  7.5000000E+00
6:                 -1.0000000E+01  1.0000000E+01
7:      2   sph     0.0000000E+00  0.0000000E+00  0.0000000E+00  9.9900000E+01
8:      3   sph     0.0000000E+00  0.0000000E+00  0.0000000E+00  1.0000000E+02
```

List 4.11 ● body example (4)

```
1:  [ B o d y ]   body example 4
2:   idbg = 0 ;  ibod = 1 ;  naz =  0
3:    non   sym  num  def
4:      1   rpp   5  -7.5000000E+00  7.5000000E+00 -7.5000000E+00  7.5000000E+00
5:                    -1.0000000E+01  1.0000000E+01
6:      2   sph   7   0.0000000E+00  0.0000000E+00  0.0000000E+00  9.9900000E+01
7:      3   sph   8   0.0000000E+00  0.0000000E+00  0.0000000E+00  1.0000000E+02
```

In the last example, body number is defined by third column, first column is skipped by the non declaration.

4.6 [R e g i o n] section

4.6.1 formats

In this section, CG geometry, importance (if necessary), volume (if necessary), and density (if necessary) are defined. Region number (num), material number (mat), region identification symbol (sym), region definition (def), volume (vol) (if necessary), importance (imp) (if necessary), and density (den) (if necessary) are required for definition. Region numbers are from 1 to 999999.

The default order is

```
num   mat  sym    def
```

You can change the order by using these symbols, but only the def must be located at the last. Mathematical expressions and user defined variable can not be used in the def. You can use the non symbol in order to skip columns.

In the case that the definition can not be written in a line, you can write in the next line without no additional symbols at the end of line. Region identification symbol can not exceed 3 characters.

If there is no density definition here, densities in [material] section is used. When density is defined here, the densities in [material] section are re-normalized. If you set density by positive value, unit is particle density[10^{24} atoms/cm^3], and negative value, mass density[g/cm^3].

As the material number, you use the material number defined in [material] section.

It is noticed that mat="-1" and "0" have special meanings. "-1" means outer void, and "0" means inner void.

4.6.2 Examples

Some examples are shown below

List 4.12 ● region example (1)

```
1:  [ R e g i o n ]
2:      1      1   tgt    +1
3:      2      2   iA5    -2 +3 +6 -7
4:      3      2   oA5    -3 +4 +6 -7
5:      4      2   iA2    -2 +3 +8 -9
6:      5      2   oA2    -3 +4 +8 -9
```

List 4.13 ● region example (2)

```
1:  [ R e g i o n ]
2:    num    mat   imp          vol          sym        def
3:      1      1   1.000000     1.000000     tgt        +1
4:      2      2   2.000000     2.000000     iA5        -2 +3 +6 -7
5:      3      2   4.000000     1.000000     oA5        -3 +4 +6 -7
6:      4      2   8.000000     1.000000     iA2        -2 +3 +8 -9
7:      5      2   16.00000     3.000000     oA2        -3 +4 +8 -9
```

List 4.14 ● region example (3)

```
1:  [ R e g i o n ]
2:    num    mat   non        non     sym        def
3:      1      1   1.000000   0.0     tgt        +1
4:      2      2   2.000000   0.0     iA5        -2 +3 +6 -7
5:      3      2   4.000000   0.0     oA5        -3 +4 +6 -7
6:      4      2   8.000000   0.0     iA2        -2 +3 +8 -9
7:      5      2   16.00000   0.0     oA2        -3 +4 +8 -9
```

Examples (1) and (3) are same definition. If you want to cancel the importance definitions in example (2), just change imp into non. When all importances are same, the importance function does not work. The importance defined in this region section affects all particles.

4.7 [C e l l] section

4.7.1 Formats

In this section, cells can be defined by surfaces described in the [surface] section. The format for the definition is based on the General Geometry (GG). You should set a cell as a closed space, and you can make a virtual space for particle transport calculation by combining the defined cells. In *PHITS* , an outer region must be explicitly defined as a cell.

Only C and $ can be used as a comment mark, but the # cannot be used as a comment mark here, since this character is used for the cell definition. File including and variable definition can be used in this section. If you want to use continuation lines, it is enough to put more than 4 blanks at the line head instead of the line sequential mark at the end of line.

The [cell] is defined in order by the data: cell number, material number, material density, cell definition, and cell parameter as keyword style. These are explained in Table 4.43. The format is shown below.

```
[ C e l l ]
cell number     mat.  number     mat.  density     cell def.      cell parameter
```

LIKE n BUT cell parameter format and repeated structure with lattice can be used. See Sec. 4.7.5 in which we describe how to use them with some examples. The cell parameters are listed and explained in Table 4.44.

Table 4.43: cell definition format

item	explanation
cell number	You can use any number from 1 to 999999.
material number	Set 0 for void, −1 for the outer region, or material number defined in [material] section.
material density	If the cell is void or the outer region, no input. When the given value is positive or negative, it is particle density [10^{24} atoms/cm^3] or mass density [g/cm^3], respectively. A material density defined in the [material] section is renormalized to the particle density given here. Thus different density materials, which have the same composition with original one, can be set in this section. A new parameter matadd is prepared in order to add different material number.
cell definition	Cell geometry is defined by both surface numbers in the [surface] section and Boolean operators, ⊔(blank)(AND), :(OR), and #(NOT). Parentheses (and) can be also used. See Sec. 4.7.2 for detail.
LIKE *n* BUT	A cell using this format is the same as the n cell, except only parameters described after BUT.
cell parameter	This format is keyword=value. As keyword, VOL(volume), TMP(temperature), TRCL(transform), U(universe), LAT(lattice), and FILL can be used. In the LIKE *n* BUT format, MAT(material) and RHO(density) can be used in addition.

Table 4.44: cell parameter

item	explanation
VOL	Volume (cm^3) of the cell is given.
TMP	Temperature (MeV) of the material in the cell is given.
TRCL	Coordinate transform for position of the cell is done using coordinate transform number defined in the [transform] section or the transform format.
U	Universe number. Number of the universe including the cell is defined. You can use any number from 1 to 999999. See Sec. 4.7.3 for detail.
LAT	Lattice number. Setting LAT=1 or 2, you can define quadratic prism or hexangular prism, respectively. See Sec. 4.7.4 for detail.
FILL	Set universe numbers to fill the cell with the universe.
MAT	This is used with LIKE *n* BUT MAT=*m* format. You can define the same cell except that its material number is *m*.
RHO	This is used with LIKE *n* BUT RHO=*x* format. You can define the same cell except that its density is *x*.

4.7.2 Description of cell definition

Cells are defined by treating regions divided by surfaces defined in the [surface] section. When you describe the definition, you need a concept, "surface sense", to make a distinction between two regions divided by the surface corresponding to an equation, $f(x,y,z) = 0$, and Boolean operators, ⊔(blank)(AND), :(OR), and #(NOT), to treat some regions.

The "surface sense" defines one region including a point (x_0, y_0, z_0), which gives $f(x_0, y_0, z_0) > 0$, as "positive sense", and the other region as "negative sense". Then, you write only the surface number in the cell definition space when you want to use a region of positive sense, and write it with minus symbol, −, when a region of negative sense. An example for this sense is shown below.

List 4.15 ● [cell] section example (1)

```
1:  [ C e l l ]
2:    1   0  -10
3:    2  -1   10
4:  [ S u r f a c e ]
5:    10  SZ  3  5
```

The 10th surface represents a sphere with a radius of 5cm. Because the inside of this sphere is negative sense, the 1st cell is defined by −10. The outer region is explicitly defined as the 2nd cell. This example gives the virtual space as shown in Fig. 4.9.

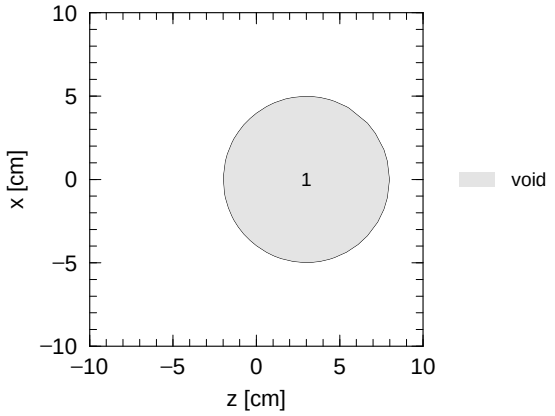


Figure 4.9: Result of the [cell] section example (1).

When you treat some regions to make the cell definition, Boolean operators are used. Symbols ␣(blank), `:`, and `#` denote intersection (AND), union (OR), and complement (NOT), respectively, as the operators. Parentheses, `(` and `)`, can be used to combine some regions. The second example in this section uses ␣(blank) and `#`.

List 4.16 ● `[cell]` section example (2)

```
 1:  [ C e l l ]
 2:    1   0  11  -12  13  -14  15  -16
 3:    2  -1  #1
 4:  [ S u r f a c e ]
 5:    11  PX  -6
 6:    12  PX   6
 7:    13  PY  -6
 8:    14  PY   6
 9:    15  PZ  -6
10:    16  PZ   6
```

In the cell definition in the 2nd line, the three numbers without minus symbol correspond to regions of positive sense of the 11th, 13th, and 15th surfaces, and those with minus correspond to regions of negative sense of the 12th, 14th, and 16th surfaces. Then, a region surrounded by the 6 surfaces is defined with ␣(blank) as the 1st cell, which is the inside of a 12cm cube. The outside of the cube is defined by the complement operator `#` as the outer region. Figure 4.10 shows the result of this example.

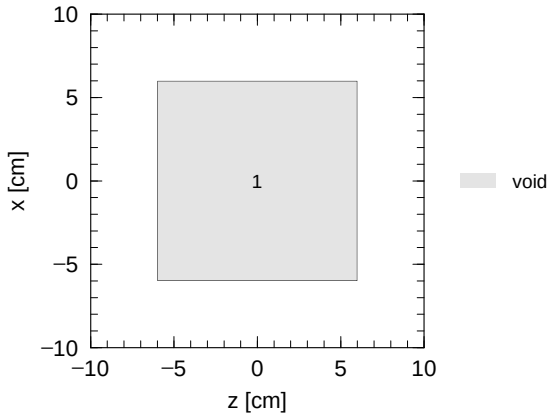


Figure 4.10: Result of the [cell] section example (2).

The next example uses `:` and parentheses. The sphere in the 1st example and the cube in the 2nd example are combined.

List 4.17 ● `[cell]` section example (3)

```
 1:  [ C e l l ]
 2:    1   0  -10 : (11  -12  13  -14  15  -16)
 3:    2  -1  #1
 4:  [ S u r f a c e ]
 5:    10  SZ   3  5
 6:    11  PX  -6
 7:    12  PX   6
 8:    13  PY  -6
 9:    14  PY   6
10:    15  PZ  -6
11:    16  PZ   6
```

A part surrounded by the parentheses in the 2nd line corresponds to the region of the 1st cell in the example (2). In this example, a region combined inside of the cube and that of the sphere in the example (1) is defined with the union operator `:` as the 1st cell. The result is shown in Fig. 4.11.

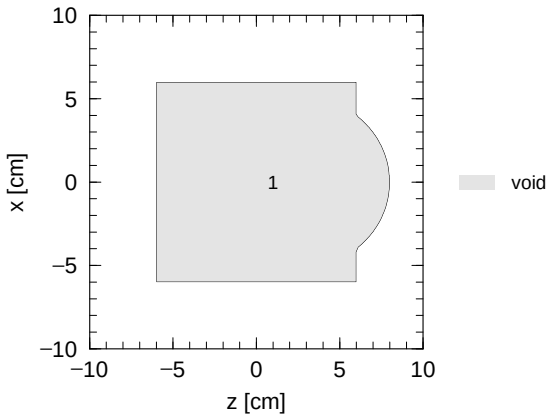


Figure 4.11: Result of the [cell] section example (3).

In the next example, division of a cube into two regions by a spherical surface is shown.

List 4.18 ● `[cell]` section example (4)

```
 1:  [ M a t e r i a l ]
 2:    mat[1]  1H 2  16O 1
 3:  [ C e l l ]
 4:    1   0       -10
 5:    2   1  1.0   10 (11  -12  13  -14  15  -16)
 6:    3  -1        #1 #2
 7:  [ S u r f a c e ]
 8:    10  SZ   3  5
 9:    11  PX  -6
10:    12  PX   6
11:    13  PY  -6
12:    14  PY   6
13:    15  PZ  -6
14:    16  PZ   6
```

This `[surface]` section is the same of the example (3). In the 5th line, the 2nd cell is defined with ␣(blank) as an overlap region between the outside of the sphere, which is the 10th surface, and the inside of the cube defined by the parentheses. The cell is filled with water defined in the `[material]` section, and its situation is shown in Fig. 4.12. The inside of the sphere is the 1st cell and void.

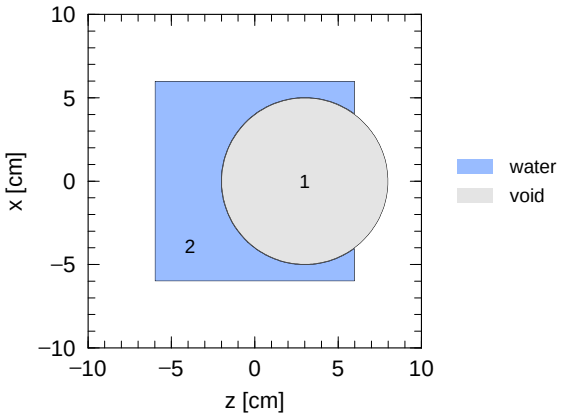


Figure 4.12: Result of the `[cell]` section example (4). The 1st and 2nd cells are filled with void and water, respectively.

4.7.3 Universe frame

In *PHITS* , you can define some universes with a cell parameter U. A region of main space for particle transport calculation is filled with a corresponding region in any universe. This function is very useful to set repeated structures introduced in Sec. 4.7.5.

An example using three spaces (one main space and two universes) shown in Fig. 4.13 is explained below. The main space includes two rectangular solids. One universe has a cylinder filled with water, and the other universe has an iron cylinder surrounded by water. The 1st cell is filled with a region of the universe 1, and the 2nd cell is filled with that of the universe 2.

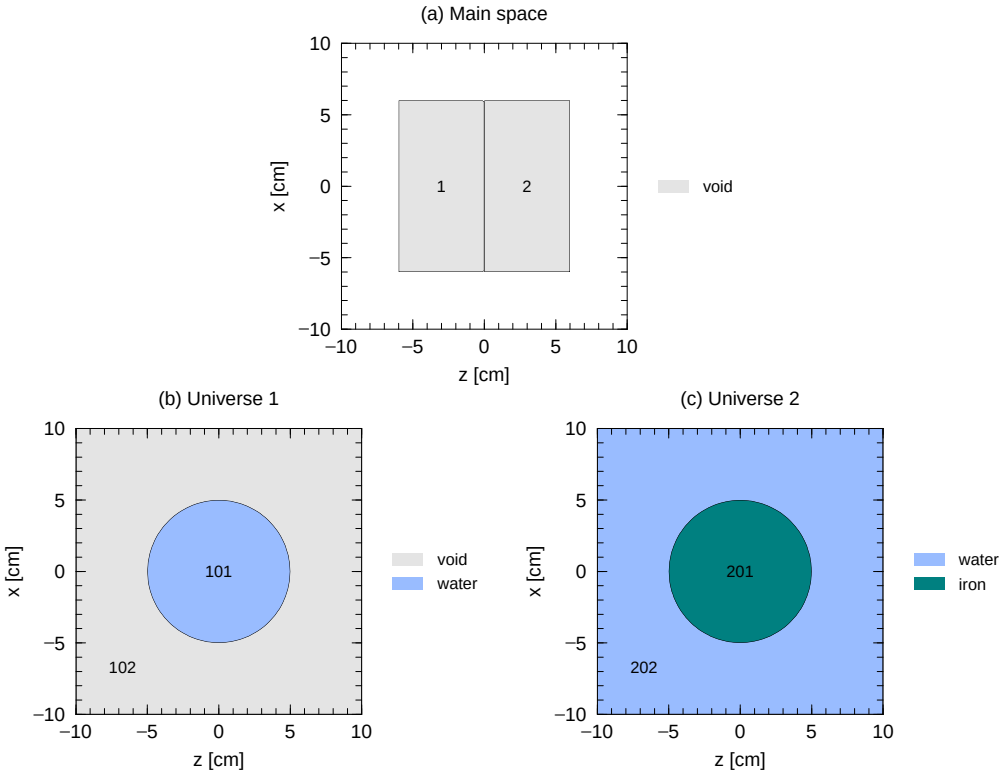


Figure 4.13: (a) Two rectangular solids. (b) Cylinder filled with water. (c) Iron cylinder in water.

List 4.19 ● `[cell]` section example (5)

```
[ M a t e r i a l ]
  mat[1]  1H 2  16O 1
  mat[2]  Fe 1
[ C e l l ]
    1  0        11  -12  13  -14  15  -17  FILL=1
    2  0        11  -12  13  -14  17  -16  FILL=2
  101  1  1.0  -10  13  -14  U=1
  102  0       #101  U=1
  201  2 10.0  -10  13  -14  U=2
  202  1  1.0  #201  U=2
    9 -1       #1 #2
[ S u r f a c e ]
  10  CY   5
  11  PX  -6
  12  PX   6
  13  PY  -6
  14  PY   6
  15  PZ  -6
  16  PZ   6
  17  PZ   0
```

The universe 1 and 2 are defined in the 7th, 8th lines and the 9th, 10th lines, respectively, using cell parameter `U`. These universes have a similar structure that a cylinder is put at the origin of the coordinate space, but their components of inside or outside of the cylinder are different from each other as shown in Fig. 4.13. In the 5th and 6th lines, the 1st and 2nd cells are, respectively, defined as regions filled with the corresponding part of the each universe using cell parameter `FILL`. The result of this example is shown in Fig. 4.14. One sees that the 1st cell consists of the 101st and 102nd cells in the universe 1, and the 2nd cell consists the 201st and 202nd cells in the universe 2.

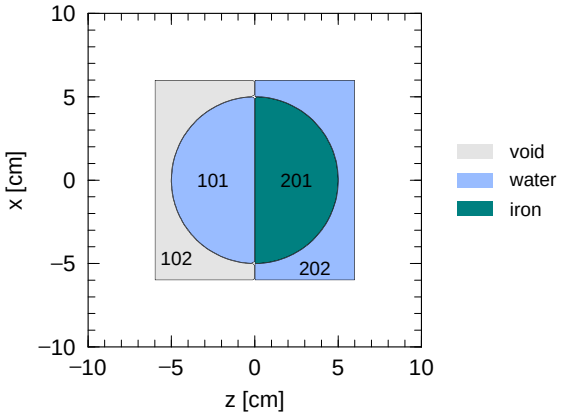


Figure 4.14: Result of the [cell] section example (5).

You cannot use an undefined region in the universe. If the 102nd cell is not defined in the 8th line as a void region, you cannot fill the 1st cell with the universe 1. In addition, you should know that all universes have the same definition for the coordinate system; position of the origin, directions of x, y, and z-axes, and scale of the space in any universe agree with those in the other universe. If the different value is used for `PX` in the 14th, 15th lines, the cube does not include a part of the cylinder as shown in Fig. 4.15.

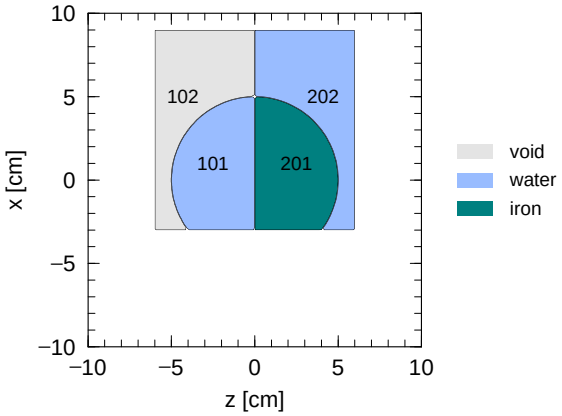


Figure 4.15: Result of the [cell] section example (5) except that the region is shifted in the x-direction.

4.7.4 Lattice definition

For making repeated structures, a cell parameter `LAT` (lattice parameter) is very useful. In this section, definition of a unit structure of the lattice and its simple usage are explained showing some examples. See Sec. 4.7.5 for more practical description.

Quadratic prism and hexangular prism shown in Fig. 4.16 can be used as a unit structure by `LAT=1` and `LAT=2`, respectively. You make one universe having the repeated structure of the lattice. Then, you fill any region with the universe. It is noted that the each unit must also be filled with another universe, which is defined with any

material or void. The numbering each component of the units in Fig. 4.16 corresponds to the order of the surface number written in the cell definition, and the lattice coordinate system, which will be explained below, depends on the order.

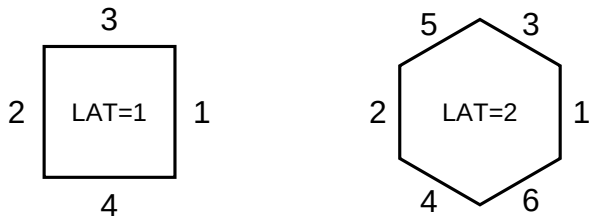


Figure 4.16: Unit structure of lattice.

An example using quadratic prism (`LAT=1`) is shown below.

List 4.20 ● `[cell]` section example (6)

```
 1:  [ M a t e r i a l ]
 2:    mat[1]  1H 2  16O 1
 3:  [ C e l l ]
 4:       1   0         11  -12   13  -14  15  -16  FILL=1
 5:     101   0        -26   25  -22   21  LAT=1  U=1  FILL=2
 6:     201   1  1.0   -90  U=2
 7:       2  -1        #1
 8:  [ S u r f a c e ]
 9:     11  PX  -6
10:     12  PX   6
11:     13  PY  -6
12:     14  PY   6
13:     15  PZ  -6
14:     16  PZ   6
15:     21  PX  -2
16:     22  PX   2
17:     23  PY  -2
18:     24  PY   2
19:     25  PZ  -2
20:     26  PZ   2
21:     90  BOX  -10 -10 -10  20 0 0  0 20 0  0 0 20
```

In the 5th line, a unit cell with `LAT=1` is defined using 4 surface numbers. Setting `U=1`, the universe 1 is defined by repeated structures of this unit, which is filled with the universe 2 defined in the 6th line. Because a cross section of the unit in the *x*-*z* plane has a square 4 cm on a side, the 1st cell defined in the 4th line as a 12 cm cube has 9-blocks as shown in Fig. 4.17. It is noted that the unit has an infinite length in the *y* direction in the universe 1 because of using only 4 surfaces. If you want to define a prism having a finite length, you have to add `-24 23` to the cell definition in the 5th line.

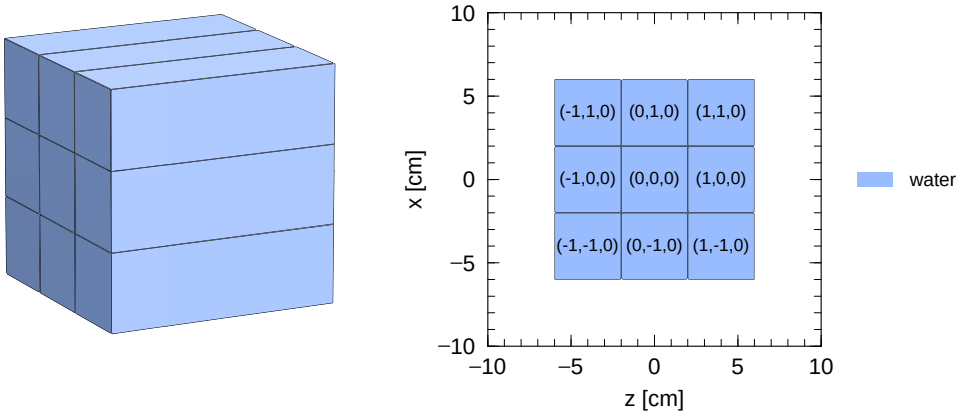


Figure 4.17: Result of the [cell] section example (6) in 3D (left) and 2D (right) images.

To distinguish cells in the repeated structure, each cell is on the lattice coordinate (s,t,u) as shown in the right panel of Fig. 4.17. Note that directions of this coordinate correspond to those of the usual coordinate (x,y,z), and are defined by the order of the surface number written in the cell definition. When you specify any cell using `mesh=reg` in tally sections, you can use the lattice and universe styles as `(201 < 101[-1 0 0] < 1)`, where the lattice coordinate is represented by `[s t u]`. See Sec. 5.1.2 for this format as well. You can see lattice coordinates by the `[t-gshow]` tally with `output=7 or 8`.

The next is an example using hexangular prism (`LAT=2`).

List 4.21 ● `[cell]` section example (7)

```
 1:  [ M a t e r i a l ]
 2:    mat[1]  1H 2  16O 1
 3:  [ C e l l ]
 4:       1   0        11 -12  13 -14  15 -16  FILL=1
 5:     101   0       -31  32 -33  34 -35  36 -24  23  LAT=2  U=1  FILL=2
 6:     201   1  1.0  -90  U=2
 7:       2  -1       #1
 8:  [ S u r f a c e ]
 9:    11  PX  -6
10:    12  PX   6
11:    13  PY  -6
12:    14  PY   6
13:    15  PZ  -6
14:    16  PZ   6
15:    23  PY  -2
16:    24  PY   2
17:    set: c1[2]
18:    31  PZ  [ c1*cos(pi/6)]
19:    32  PZ  [-c1*cos(pi/6)]
20:    33  P   1  0  [ 1/tan(pi/3)]  [ c1]
21:    34  P   1  0  [ 1/tan(pi/3)]  [-c1]
22:    35  P   1  0  [-1/tan(pi/3)]  [ c1]
23:    36  P   1  0  [-1/tan(pi/3)]  [-c1]
24:    90  BOX  -10 -10 -10  20 0 0  0 20 0  0 0 20
```

A hexagon with `LAT=2` is defined in the 5th line using 6 surfaces defined in the 17th-23th lines. The hexagonal prism is restricted in the y-direction by `-24 23` in the cell definition, and is filled with the universe 2, namely, water as written in the 6th line. The 1st cell has the repeated structure defined as the universe 1. Figure 4.18 shows the result of this example. One can see that some prisms near edges of the 1st cell, which is defined as a 12 cm cube, are only partly used. Directions of the lattice coordinate shown in the right panel depend on the order of the surface number written in the cell definition. When you specify any cell using `mesh=reg` in tally sections, you can use the lattice and universe styles as `(201 < 101[-2 0 0] < 1)`, where the lattice coordinate is represented by `[s t u]`. See Sec. 5.1.2 for this format as well. You can see lattice coordinates by the `[t-gshow]` tally with `output=7 or 8`.

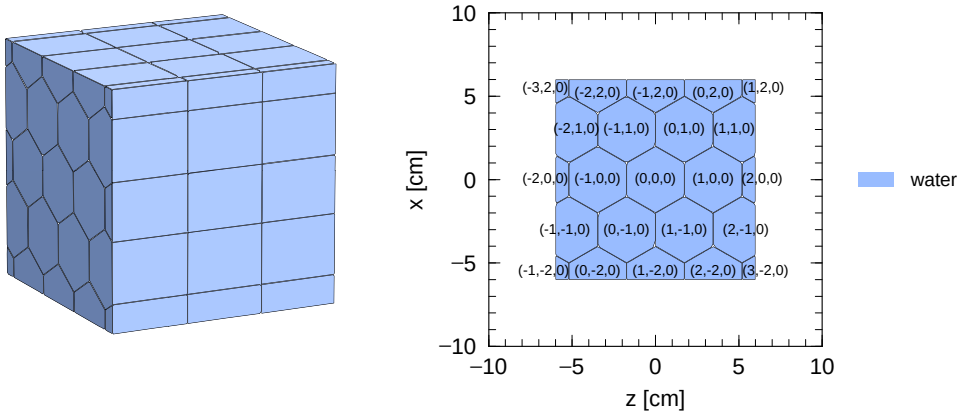


Figure 4.18: Result of the [cell] section example (7) in 3D (left) and 2D (right) images.

4.7.5 Repeated structures

You can use some simple procedures in *PHITS* to make repeated structures, where the same or similar units are put repeatedly. Using a lattice parameter explained in Sec. 4.7.4 is one of them, and another is the `LIKE n BUT cell parameter` format.

`LIKE n BUT` cell parameter

Using this format, you can make a little different cell from original one. Only elements corresponding to cell parameters written after `BUT` are different from the *n* cell. Cell parameters that can be used in this format are shown in Table 4.44. In the following example, two cell parameters `TRCL` and `MAT` are used.

List 4.22 ● `[cell]` section example (8)

```
 1:  [ M a t e r i a l ]
 2:    mat[1]  1H 2  16O 1
 3:    mat[2]  Fe 1
 4:  [ C e l l ]
 5:      1   0       -10  13 -14  #2 #3 #4
 6:      2   1  1.0   11 -12  13 -14  15 -16
 7:      3   LIKE 2 BUT  TRCL=1
 8:      4   LIKE 2 BUT  TRCL=2  MAT=2
 9:      5  -1        #(-10  13  -14)
10:  [ S u r f a c e ]
11:    10  CY   10
12:    11  PX  -2
13:    12  PX   2
14:    13  PY  -2
15:    14  PY   2
16:    15  PZ  -2
17:    16  PZ   2
18:  [T r a n s f o r m ]
19:  *tr1   3 0 -5
20:  *tr2   0 0  6  30 90 120  90 0 90  60 90 30  1
```

A 4 cm cube filled with water is defined in the 6th line, and is put at the origin of the coordinate system. Inside of this cube is the 2nd cell regarded as the original cell in this example. In the 7th and 8th lines, respectively, the 3rd and 4th cells are defined with the `LIKE n BUT` format, where $n = 2$. Figure 4.19 shows the result of the example. The coordinate system of the 3rd cell is transformed using the cell parameter `TRCL=1`, where the coordinate transform number 1 is defined in the 19th line in the `[transform]` section. That of the 4th cell is also transformed with `TRCL=2`. Moreover, the material inside of the cell is replaced with iron defined as the material number 2 in the 3rd line.

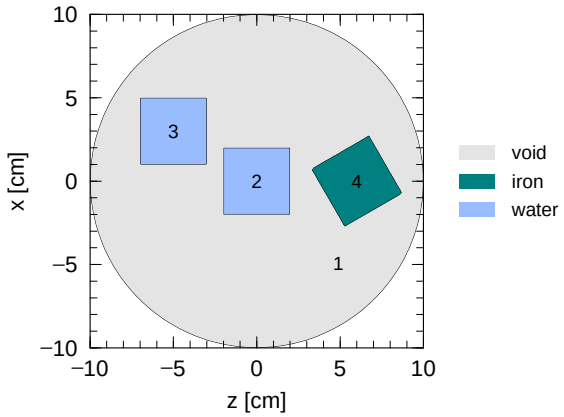


Figure 4.19: Result of the [cell] section example (8).

Nesting structure with lattice

A nesting structure can be used on the basis of universe frame in Sec. 4.7.3. For example, the universe 1 is filled with the universe 2, and the universe 2 is filled with the universe 3. Moreover, the 3rd can be also filled with another universe. Then, you can define the nesting structure. The maximum number of the nesting level is 10, which corresponds to a parameter `mxlv` given in a file `param.inc`.

In the next example, there are nine square poles defined with `LAT=1`, and three of these have a different structure from the others.

List 4.23 ● `[cell]` section example (9)

```
 1:  [ M a t e r i a l ]
 2:    mat[1]  1H 2  16O 1
 3:    mat[2]  Fe 1
 4:  [ C e l l ]
 5:       1   0         11  -12   13  -14  15  -16  FILL=1
 6:    101   0        -26   25  -22   21  LAT=1  U=1
 7:                   FILL=-1:1  -1:1  0:0
 8:                   2 2 3  2 3 2  3 2 2
 9:    201   1  1.0   -90  U=2
10:    301   2 10.0   -10  U=3
11:    302   0         10  U=3
12:      2  -1        #1
13:  [ S u r f a c e ]
14:    10  CY   1.5
15:    11  PX  -6
16:    12  PX   6
17:    13  PY  -6
18:    14  PY   6
19:    15  PZ  -6
20:    16  PZ   6
21:    21  PX  -2
22:    22  PX   2
23:    25  PZ  -2
24:    26  PZ   2
25:    90  BOX  -10 -10 -10  20 0 0  0 20 0  0 0 20
```

Definition of the 1st cell and the unit of lattice in the 5th and 6th lines, respectively, is the same of that in the `[cell]` section example (6). However, a format of the cell parameter `FILL` written in the 7th and 8th lines is different. In the 7th line, regions treated in this calculation are given in the lattice coordinate system. The numbers in the next line correspond to the universe number filling each lattice at (s,t,u), where the order is $(-1,-1,0),(0,-1,0),(1,-1,0),(-1,0,0),\ldots,(1,1,0)$; i.e., a lattice at $(-1,-1,0)$ is filled with the universe 2 and that at $(1,-1,0)$ is filled with the universe 3. The universe 2 is defined in the 9th line as space filled with water. On the other hand, the universe 3 defined in the 10th and 11th lines has an iron cylinder at the origin. The result of this example is shown in Fig. 4.20. One can see that three lattices at $(1,-1,0),(0,0,0)$, and $(-1,1,0)$ have the iron cylinder. When you specify any cell using `mesh=reg` in tally sections, you can use the lattice and universe styles as `(302 < 101[0 0 0] < 1)`, where the lattice coordinate is represented by `[s t u]`. See also Sec. 5.1.2 for this format.

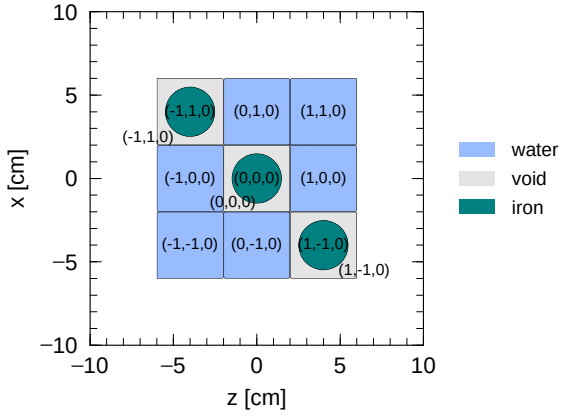


Figure 4.20: Result of the [cell] section example (9).

More complex example is shown below.

List 4.24 ● `[cell]` section example (10)

```
 1:  [ M a t e r i a l ]
 2:    mat[1]  1H 2  16O 1
 3:    mat[2]  Fe 1
 4:  [ C e l l ]
 5:      1   0        11  -12   13  -14  15  -16  FILL=1
 6:    101   0       -26   25  -22   21  LAT=1  U=1
 7:                  FILL=-1:1  -1:1  0:0
 8:                  2 2 3(1 0 1)  2 3(1 0 1) 2  3(1 0 1) 2 2
 9:    201   1  1.0  -90  U=2
10:    301   0       -36   35  -32   31  LAT=1  U=3
11:                  FILL=-1:0  -1:0  0:0
12:                  4 2  2 4
13:    401   2 10.0  -10  U=4
14:    402   0        10  U=4
15:      2  -1       #1
16:  [ S u r f a c e ]
17:    10  CY   0.5
18:    11  PX  -6
19:    12  PX   6
20:    13  PY  -6
21:    14  PY   6
22:    15  PZ  -6
23:    16  PZ   6
24:    21  PX  -2
25:    22  PX   2
26:    25  PZ  -2
27:    26  PZ   2
28:    31  PX  -1
29:    32  PX   1
30:    35  PZ  -1
31:    36  PZ   1
32:    90  BOX  -10 -10 -10  20 0 0  0 20 0  0 0 20
```

The virtual space made by this input is shown in Fig. 4.21. The nine square poles are defined with the lattice parameter. Furthermore, three of these consist of 4 units of the other lattice. The `(1 0 1)` in the 8th line denotes the transformation of the coordinate system that the origin is shifted by 1cm in the *x*- and *z*-direction. When you specify any cell using `mesh=reg` in tally sections, you can use the lattice and universe styles as `(402 < 301[-1 -1 0] < 101[0 0 0] < 1)`, where the lattice coordinate is represented by `[s t u]`. See also Sec. 5.1.2 for this format.

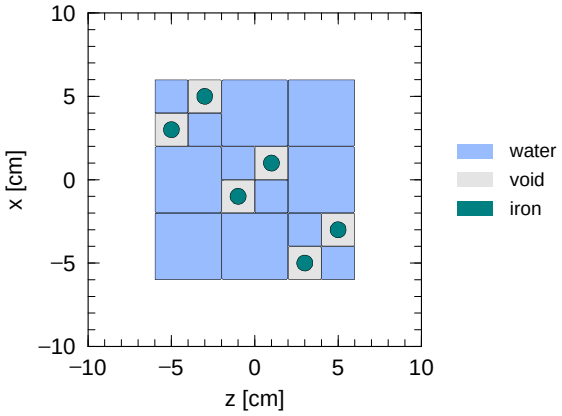


Figure 4.21: Result of the [cell] section example (10).

Voxel phantom

In *PHITS* , you can make a virtual space using voxel phantom for calculation on complex structures, such as the human body or organism. First, a little cube is defined as a unit of the lattice with `LAT=1`. Second, you set a repeated structure of a large size using the unit. Third, you fill each unit with any universe, which is itself filled with biological matter, such as compounds of carbon and water.

In an example below, a 10 cm cube consisting of 2cm cubes (voxels) of $5 \times 5 \times 5 = 125$ is described.

List 4.25 ● `[cell]` section example (11)

```
 1:  [ M a t e r i a l ]
 2:    mat[1]  1H 2  16O 1
 3:    mat[2]  Fe 1
 4:  [ C e l l ]
 5:       1   0   11  -12   13  -14  15  -16  FILL=1
 6:    101   0  -20  LAT=1  U=1
 7:              FILL=-2:2  -2:2 -2:2
 8:              2 2 2 2 2  2 2 2 2 2  2 2 3 2 2  2 2 2 2 2  2 2 2 2 2
 9:              2 2 2 2 2  2 3 3 2 2  2 3 4 3 2  2 3 3 2 2  2 2 2 2 2
10:              2 2 2 2 2  2 3 3 3 2  3 4 4 4 3  2 3 3 3 2  2 2 2 2 2
11:              2 2 2 2 2  2 2 3 3 2  2 3 4 3 2  2 2 3 3 2  2 2 2 2 2
12:              2 2 2 2 2  2 2 2 2 2  2 2 3 2 2  2 2 2 2 2  2 2 2 2 2
13:    201   0       -90  U=2
14:    301   2 10.0  -90  U=3
15:    401   1  1.0  -90  U=4
16:      2  -1       #1
17:  [ S u r f a c e ]
18:    11  PX  -5
19:    12  PX   5
20:    13  PY  -5
21:    14  PY   5
22:    15  PZ  -5
23:    16  PZ   5
24:    20  BOX  -1 -1 -1  2 0 0  0 2 0  0 0 2
25:    90  BOX  -10 -10 -10  20 0 0  0 20 0  0 0 20
```

As a unit of voxel, a 2 cm cube is defined in the 24th line. Furthermore, the 1st cell that is inside of a 10 cm cube has a repeated structure through definition in the 5th line. The region of the lattice coordinate space is determined in the 7th line. The order of voxel in the 8th-12th lines is as follows: $(-2,-2,-2),(-1,-2,-2),\ldots,(2,2,2)$ represented by the lattice coordinate. `2` in the 8th-12th lines means the universe 2, which is void, and `3` and `4`, namely the universe 3 and 4, correspond to iron and water, respectively. Figure 4.22 represents the result of this example that is a distorted iron box including water its inside. When you specify any cell using `mesh=reg` in tally sections, you can use the lattice and universe styles as `(401 < 101[0 0 0] < 1)`, where the lattice coordinate is represented by `[s t u]`. See also Sec. 5.1.2 for this format. It is noted that you cannot use formats as `(301 < 101[-2:2 -2:2 -2:2] < 1)` because not all the `101[-2:2 -2:2 -2:2]` cell have the 301st cell.

Figure 4.22: Results of the [cell] section example (11) in 3D images. The structure in the right panel is removed its iron surface from the original one in the left panel.

For time shortening, you can use `ivoxel` in the `[parameters]` section. When you perform *PHITS* calculation with `ivoxel=2`, voxel data are output in `file(18)` in binary and then the calculation is stopped. From the next calculation with `ivoxel=1`, a process of the data output is omitted and the calculation time is shortened. If you use a very huge voxel data, using `infl` may become to be convenient.

4.8 [S u r f a c e] section

4.8.1 Formats

Surface is defined in this section. Only C and $ can be used as a comment mark, but # can not be used as a comment mark here. The file including and variable definition can be set in this section. If you want to use continuation lines, it is enough to put more than 4 blanks at the line head instead of the line sequential mark at the end of line.

The order of format is, surface number, coordinate transform number, surface symbol, and surface definition. You can use mathematical expressions and user defined variables in the surface definition. Surface definition by macro body can be used. Surface number is limited from 1 to 999999.

Formats and examples are shown below

```
[ S u r f a c e ]
surface number     transform number     surface symbol     surface definition
```

Table 4.45: surface definition format

item	explanation
surface number	1 ~ 999999
coordinate transform number	if no coordinate transform, no input, else with coordinate transform, use number n of TRn in [transform] section
surface symbol	surface symbol in surface card list, or symbol of macro body
surface definition	1 ~ 15 inputs depends on surfaces

4.8.2 Examples

List 4.26 ● [surface] section example (1)

```
 1:  [surface]
 2:    1   cz     5.0
 3:    2   cz    10.0
 4:    3   cz    15.0
 5:    4   cz    20.0
 6:    5   pz     0.0
 7:    6   pz     5.0
 8:    7   pz    10.0
 9:    8   pz    15.0
10:    9   pz    55.0
11:   10   pz    60.0
```

Table 4.46: surface card

surface symbol	type	explanation	equation	input numeric value
P	plane	multi-purpose	$Ax + By + Cz - D = 0$	$A\ B\ C\ D$
PX		vertical with X-axis	$x - D = 0$	D
PY		vertical with Y-axis	$y - D = 0$	D
PZ		vertical with Z-axis	$z - D = 0$	D
SO	sphere	origin is center	$x^2 + y^2 + z^2 - R^2 = 0$	R
S		multi-purpose	$(x - \bar{x})^2 + (y - \bar{y})^2 + (z - \bar{z})^2 - R^2 = 0$	$\bar{x}\ \bar{y}\ \bar{z}\ R$
SX		center on X-axis	$(x - \bar{x})^2 + y^2 + z^2 - R^2 = 0$	$\bar{x}\ R$
SY		center on Y-axis	$x^2 + (y - \bar{y})^2 + z^2 - R^2 = 0$	$\bar{y}\ R$
SZ		center on Z-axis	$x^2 + y^2 + (z - \bar{z})^2 - R^2 = 0$	$\bar{z}\ R$
C/X	cylinder	parallel with X-axis	$(y - \bar{y})^2 + (z - \bar{z})^2 - R^2 = 0$	$\bar{y}\ \bar{z}\ R$
C/Y		parallel with Y-axis	$(x - \bar{x})^2 + (z - \bar{z})^2 - R^2 = 0$	$\bar{x}\ \bar{z}\ R$
C/Z		parallel with Z-axis	$(x - \bar{x})^2 + (y - \bar{y})^2 - R^2 = 0$	$\bar{x}\ \bar{y}\ R$
CX		on X-axis	$y^2 + z^2 - R^2 = 0$	R
CY		on Y-axis	$x^2 + z^2 - R^2 = 0$	R
CZ		on Z-axis	$x^2 + y^2 - R^2 = 0$	R
K/X	cone	parallel with X-axis	$\sqrt{(y - \bar{y})^2 + (z - \bar{z})^2} - t(x - \bar{x}) = 0$	$\bar{x}\ \bar{y}\ \bar{z}\ t^2\ \pm 1$
K/Y		parallel with Y-axis	$\sqrt{(x - \bar{x})^2 + (z - \bar{z})^2} - t(y - \bar{y}) = 0$	$\bar{x}\ \bar{y}\ \bar{z}\ t^2\ \pm 1$
K/Z		parallel with Z-axis	$\sqrt{(x - \bar{x})^2 + (y - \bar{y})^2} - t(z - \bar{z}) = 0$	$\bar{x}\ \bar{y}\ \bar{z}\ t^2\ \pm 1$
KX		on X-axis	$\sqrt{y^2 + z^2} - t(x - \bar{x}) = 0$	$\bar{x}\ t^2\ \pm 1$
KY		on Y-axis	$\sqrt{x^2 + z^2} - t(y - \bar{y}) = 0$	$\bar{y}\ t^2\ \pm 1$
KZ		on Z-axis	$\sqrt{x^2 + y^2} - t(z - \bar{z}) = 0$	$\bar{z}\ t^2\ \pm 1$
				± 1 is only needed for 1 sheet code
SQ	ellipse hyperboloid paraboloid	parallel with X-, Y-, or Z- axis	$A(x - \bar{x})^2 + B(y - \bar{y})^2 + C(z - \bar{z})^2 + 2D(x - \bar{x}) + 2E(y - \bar{y}) + 2F(z - \bar{z}) + G = 0$	$A\ B\ C\ D\ E$ $F\ G\ \bar{x}\ \bar{y}\ \bar{z}$
GQ	cylinder code ellipse hyperboloid paraboloid	non parallel with X-, Y- and Z-axis	$Ax^2 + By^2 + Cz^2 + Dxy + Eyz + Fzx + Gx + Hy + Jz + K = 0$	$A\ B\ C\ D\ E$ $F\ G\ H\ J\ K$
TX	ellipse torus torus	parallel with X-, Y-, or Z-axis	$(x - \bar{x})^2/B^2 + (\sqrt{(y - \bar{y})^2 + (z - \bar{z})^2} - A)^2/C^2 - 1 = 0$	$\bar{x}\ \bar{y}\ \bar{z}\ A\ B\ C$
TY			$(y - \bar{y})^2/B^2 + (\sqrt{(x - \bar{x})^2 + (z - \bar{z})^2} - A)^2/C^2 - 1 = 0$	$\bar{x}\ \bar{y}\ \bar{z}\ A\ B\ C$
TZ			$(z - \bar{z})^2/B^2 + (\sqrt{(x - \bar{x})^2 + (y - \bar{y})^2} - A)^2/C^2 - 1 = 0$	$\bar{x}\ \bar{y}\ \bar{z}\ A\ B\ C$
X Y Z P		defined by points		

The cone defined by $\bar{x}, \bar{y}$, or $\bar{z}$ has two sheets as the center being the coordinate of the top along the direction of each axis. If you set to be 1 for ± 1, the upper sheet is used, and the lower sheet is used in the case of -1. When the value is not given, both sheets are used.

4.8.3 Macro body

Table 4.47: macro body card

symbol	type	numerical input	explanation
BOX	optional BOX (all angles are 90°)	$V_x\ V_y\ V_z$	base point coordinate
		$A1x\ A1y\ A1z$	vector from base point to first surface
		$A2x\ A2y\ A2z$	vector from base point to first surface
		$A3x\ A3y\ A3z$	vector from base point to third surface
RPP	rectangular solid (each surface is vertical with xyz)	$X_{min}\ X_{max}$	minimum x and maximum x
		$Y_{min}\ Y_{max}$	minimum y and maximum y
		$Z_{min}\ Z_{max}$	minimum z and maximum z
SPH	sphere (same with general sphere S)	$V_x\ V_y\ V_z$	center coordinate
		R	radius
RCC	cylinder	$V_x\ V_y\ V_z$	center coordinate of bottom face
		$H_x\ H_y\ H_z$	vector from $V_x\ V_y\ V_z$ to center coordinate of top face
		R	radius
RHP or HEX	optional hexangular prism prism	$v1\ v2\ v3$	base point coordinate
		$h1\ h2\ h3$	height vector from base point
		$r1\ r2\ r3$	vector from base point to first surface
		$s1\ s2\ s3$	vector from base point to second surface
		$t1\ t2\ t3$	vector from base point to third surface
REC	right elliptical cylinder	$V_x\ V_y\ V_z$	center coordinate of bottom face
		$H_x\ H_y\ H_z$	vector from $V_x\ V_y\ V_z$ to center coordinate of top face
		$V_{1x}\ V_{1y}\ V_{1z}$	major axis vector of ellipse orthogonal to $\vec{H}$
		$V_{2x}\ V_{2y}\ V_{2z}$	minor axis vector of ellipse orthogonal to $\vec{H}$ and $\vec{V}_1$
TRC	truncated right-angle cone	$V_x\ V_y\ V_z$	center coordinate of bottom face of cone
		$H_x\ H_y\ H_z$	height vector from $V_x\ V_y\ V_z$
		R_1	radius of bottom face
		R_2	radius of top face
ELL	ellipsoid	If $Rm > 0$,	
		$V_{1x}\ V_{1y}\ V_{1z}$	coordinate of the first focus
		$V_{2x}\ V_{2y}\ V_{2z}$	coordinate of the second focus
		Rm	major axis length
		If $Rm < 0$,	
		$V_{1x}\ V_{1y}\ V_{1z}$	center coordinate of ellipsoid
		$V_{2x}\ V_{2y}\ V_{2z}$	major axis vector
		Rm	minor axis length
WED	wedge	$V_x\ V_y\ V_z$	coordinate of the top
		$V_{1x}\ V_{1y}\ V_{1z}$	vector to the first triangle face
		$V_{2x}\ V_{2y}\ V_{2z}$	vector to the second triangle face
		$V_{3x}\ V_{3y}\ V_{3z}$	height vector

4.8.4 Examples

List 4.27 ● `[surface]` section example (2) corresponding the `[cell]` section example (2)

```
1:  [surface]
2:    1   rpp -15 15 -5 5  0 60
3:    2   rpp  -5  5 -5 5  0 20
4:    4   rpp -15 15 -5 5  0 20
5:    5   rpp -20 20 -5 5  0 40
6:    6   rpp -20 20 -5 5  0 20
7:    7   rpp -20 20 -5 5 40 60
8:    3   c/y   0 10  4
```

4.8.5 Surface definition by macro body

When you use a surface defined by a macro body in the cell definition, "-" means inside of the macro body and "+" means outside of the macro body. Each surface composing a macro body, can be used in the cell definition. In that case, you should write macro body number with "." and surface number. Surface number is shown below.

Table 4.48: surface number in macro body

symbol	surface number	explanation
BOX	1	surface vertical with the end of $A1x$ $A1y$ $A1z$
	2	surface vertical with the origin of $A1x$ $A1y$ $A1z$
	3	surface vertical with the end of $A2x$ $A2y$ $A2z$
	4	surface vertical with the origin of $A2x$ $A2y$ $A2z$
	5	surface vertical with the end of $A3x$ $A3y$ $A3z$
	6	surface vertical with the origin of $A3x$ $A3y$ $A3z$
RPP	1	surface at X_{max}
	2	surface at X_{min}
	3	surface at Y_{max}
	4	surface at Y_{min}
	5	surface at Z_{max}
	6	surface at Z_{min}
SPH		sphere surface
RCC	1	side face of cylinder
	2	surface vertical with the end of H_x H_y H_z
	3	surface vertical with the origin of H_x H_y H_z
RHP or HEX	1	surface vertical with the end of $r1$ $r2$ $r3$
	2	opposite face for surface 1
	3	surface vertical with the end of $s1$ $s2$ $s3$
	4	opposite face for surface 3
	5	surface vertical with the end of $t1$ $t2$ $t3$
	6	opposite face for surface 5
	7	surface vertical with the end of $h1$ $h2$ $h3$
	8	surface vertical with the origin of $h1$ $h2$ $h3$
REC	1	side face of cylinder
	2	surface vertical with the end of H_x H_y H_z
	3	surface vertical with the origin of H_x H_y H_z
TRC	1	side face of cone
	2	surface vertical with the end of H_x H_y H_z
	3	surface vertical with the origin of H_x H_y H_z
ELL		This is defined as normal surface.
WED	1	surface including top and bottom hypotenuses
	2	surface including $\vec{V}_2$ and $\vec{V}_3$
	3	surface including $\vec{V}_1$ and $\vec{V}_3$
	4	surface including $\vec{V}_1$, $\vec{V}_2$ and the end of $\vec{V}_3$
	5	surface including $\vec{V}_1$, $\vec{V}_2$ and the origin of $\vec{V}_3$

4.9 [T r a n s f o r m] section

4.9.1 Formats

You can define the coordinate transform in this section. Only `C` and `$` can be used as a comment mark. File including and variable definition can be set in this section.

The coordinate transformation defined in this section can be used in `[source]` section, `[surface]` section, `r-z, xyz` mesh of tally and the magnetic field.

Formats and examples are shown below.

```
[ T r a n s f o r m ]
   TRn    O1   O2   O3   B1   B2   B3   B4   B5   B6   B7   B8   B9   M
```

Table 4.49: transform definition

item	explanation
n	transform number 1 ~ 999 $*TRn$ means that B_i is not a cosine but an angle.(degree)
O_1 O_2 O_3	transposition vector
$B_1 \sim B_9$	rotation matrix
M	= 1 means that transposition vector is in sub coordinate system defined in main coordinate system. = −1 means that transposition vector is in main coordinate system defined in sub coordinate system.

Default values are shown below.

```
   TRn    0   0   0    1   0   0   0   1   0   0   0   1    1
```

4.9.2 Mathematical definition of the transform

The mathematical definition in terms of transposition vector and rotation matrix is the following,
In the case of $M = 1$,

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} B_1 & B_2 & B_3 \\ B_4 & B_5 & B_6 \\ B_7 & B_8 & B_9 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} + \begin{pmatrix} O_1 \\ O_2 \\ O_3 \end{pmatrix}$$

In the case of $M = -1$,

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} B_1 & B_2 & B_3 \\ B_4 & B_5 & B_6 \\ B_7 & B_8 & B_9 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} + \begin{pmatrix} B_1 & B_4 & B_7 \\ B_2 & B_5 & B_8 \\ B_3 & B_6 & B_9 \end{pmatrix} \begin{pmatrix} O_1 \\ O_2 \\ O_3 \end{pmatrix}$$

Here,

$$
\begin{aligned}
B_1 &= \cos(x', x) \\
B_2 &= \cos(x', y) \\
B_3 &= \cos(x', z) \\
B_4 &= \cos(y', x) \\
B_5 &= \cos(y', y) \\
B_6 &= \cos(y', z) \\
B_7 &= \cos(z', x) \\
B_8 &= \cos(z', y) \\
B_9 &= \cos(z', z)
\end{aligned}
$$

4.9.3 Examples (1)

List 4.28 ● `[transform]` section example (1)

```
1:  [ T r a n s f o r m ]
2:  *tr1       0.0000000E+00  0.0000000E+00  1.4000000E+03
3:             1.3500000E+02  9.0000000E+01  4.5000000E+01
4:             9.0000000E+01  0.0000000E+00  9.0000000E+01
5:             2.2500000E+02  9.0000000E+01  1.3500000E+02    1
6:  *tr2       0.0000000E+00  0.0000000E+00  2.5800000E+03
7:             3.0000000E+02  9.0000000E+01  2.1000000E+02
8:             9.0000000E+01  0.0000000E+00  9.0000000E+01
9:             3.0000000E+01  9.0000000E+01  3.0000000E+02    1
```

In this example, `tr1` rotates the coordinate by 135 degrees around y axis, and transports 140 cm to z direction, while `tr2` rotates 300 degrees around y axis, and transports 258 cm to z direction. Because of `TRn` with $*$, you can give angles (in units of degree) directly for $B_i, (i = 1, \ldots, 9)$.

4.9.4 Examples (2)

List 4.29 ● `[transform]` section example (2)

```
 1:  [ T r a n s f o r m ]
 2:  set: c10[90]  $ angle of around Z (degree)
 3:  set: c20[30]  $ angle of around Y (degree)
 4:  set: c30[0]   $ angle of around X (degree)
 5:
 6:  tr1   0 0 0
 7:        cos(c10/180*pi)*cos(c20/180*pi)
 8:        sin(c10/180*pi)*cos(c30/180*pi)+cos(c10/180*pi)*sin(c20/180*pi)*sin(c30/180*pi)
 9:        sin(c10/180*pi)*sin(c30/180*pi)-cos(c10/180*pi)*sin(c20/180*pi)*cos(c30/180*pi)
10:       -sin(c10/180*pi)*cos(c20/180*pi)
11:        cos(c10/180*pi)*cos(c30/180*pi)-sin(c10/180*pi)*sin(c20/180*pi)*sin(c30/180*pi)
12:        cos(c10/180*pi)*sin(c30/180*pi)+sin(c10/180*pi)*sin(c20/180*pi)*cos(c30/180*pi)
13:        sin(c20/180*pi)
14:       -cos(c20/180*pi)*sin(c30/180*pi)
15:        cos(c20/180*pi)*cos(c30/180*pi)
16:        1
```

In this example, tr1 rotates the coordinate by c10 degree around z axis, c20 degree around y axis and finally c30 degree around x axis. You can set c10, c20, c30, and rotate the coodinate to any direction as you want.

4.10 [I m p o r t a n c e] section

The importance for CG region and GG cell can be defined in this section. The importance can be also defined in [region] section , but defined importance in this section has high priority than ones in [region]. The [region] importances are ignored if the importance is defined in both. If the importance is not defined, it is set as "1.0". In the [region] section, the importance is defined only for all particles, but you can set importance for individual particle by this section.

Maximum 6 [importance] sections are allowed to be defined in a input file.

```
      [ I m p o r t a n c e ]
         part =  proton  neutron
          reg               imp
            1              1.000000
           11              5.000000
    ( { 2 - 5 } 8 9 )      2.000000
       ( 11 12 15 )        3.000000
    ( 6<10[1 0 0]<u=3 )    6.000000
           ....            ........
           ....            ........
```

Particle is defined as `part =` at the first line. If the "part" is not defined, default value is defined as `part = all`. The format to describe particles is the same as in tally definition. However, it can distinguish ityp only, each nucleus is not specified.

If you want to change the order of region number (`reg`) and (`imp`), set as "`imp reg`". You can use the skip operator `non`. Even if you use the GG, you should write not `cell` but `reg` here.

You can use the format like `( { 2 - 5 } 8 9 )`, and you can use the lattice and universe style as `( 6 < 10[1 0 0] < u=3 )`. But you need to close a value by `( )` if it is not a single numeric value.

The importance of bottom level is a product by each importance at each level. In *PHITS* , importance of a specific cell at bottom level can be defined by above format. By using the format, we can define different importance into each lattice. If the importance is double-defined, the first defined importance is valid.

If you set large importance to particles which have strong penetration through matter such as neutrino, *PHITS* calculation takes time too much. If you define part=all, neutrino is included. You must give attention about it.

Some rules can be used to define an importance of a cell in a repeated structures and lattices. For example, cells 5, 6, and 7 on a bottom level are included by cells 11, 12, and 13 on upper level, we can define the importance as

```
1:   [ Importance ]
2:      reg            imp
3:    ( 5 6 7 < 11 )   2.0
4:    ( 5 6 7 < 12 )   4.0
5:    ( 5 6 7 < 13 )   8.0
6:    ( 11 12 13 )     1.0
```

or

```
1:   [ Importance ]
2:      reg            imp
3:    ( 5 6 7 )        1.0
4:      11             2.0
5:      12             4.0
6:      13             8.0
```

Above two definitions give same results, but in the latter case, the importance for cells 5, 6, and 7 are displayed as 1.0 at the importance summary.

4.11 [Weight Window] section

The weight window function can be defined in this section. Maximum 6 [weight window] sections are allowed to be defined in a input file.

```
      [ Weight Window ]
         part =  proton  neutron
          eng =  5
        ( tim =  5 )
                 6.00e-7  3.98e-1  1.00e+0  7.00e+0  5.00e+4
          reg                ww1        ww2        ww3
            1              0.010000   0.100000   0.001000
           11              0.005000   0.050000   0.000300
    ( { 2 - 5 } 8 9 )      0.001000   0.010000   0.000100
       ( 11 12 15 )        0.000500   0.005000   0.000030
    ( 6<10[1 0 0]<u=3 )    0.000010   0.001000   0.000010
           ....            ........   ........   ........
                             ww4        ww5
                           0.010000   0.100000
                           0.005000   0.050000
                           0.001000   0.010000
                           0.000500   0.005000
                           0.000010   0.001000
                           ........   ........
```

Particle is defined in the first line as `part = .` `part = all` means all particles. The format to describe particles `part =` is the same format as in tally definition. However, it can distinguish ityp only, each nucleus is not specified.

Next you define the energy mesh or time mesh. First, you define the number of mesh by `eng =` or `tim =` and, in next line, the values of each mesh ($e_1, e_2, e_3, ...$). Minimum value of weight window for each mesh should be defined in the followings. Each minimum values are like ww1, ww2, ww3, ... where wwi is a window minimum value for a mesh $e_{i-1} < E < e_i$. $e_0 = 0$ and $t_0 = -\infty$ is assumed. If there exists no `eng =` / `tim =` definitions, energy / time mesh are not prepared. In this case, you should set only ww1.

Region number (`ref`) must be written at the first column. As above example, you can make another table for wwi definitions. From second table, the region definition can be skipped as the example. You can use the skip operator `non` in this section. Even if you use GG, you should write the symbol not `cell` but `reg` in the section.

You can use the format `( { 2 - 5 } 8 9 )`, and you can use the lattice and universe style as `( 6 < 10[1 0 0] < u=3 )`. But you need to close a value by `(  )` if it is not a single numeric value.

If you set large weight window to particles which has strong penetration through matter such as neutrino, *PHITS* calculation takes time too much. If you define part=all, neutrino is included. You must give attention about it.

4.12 [V o l u m e] section

Volume for CG region and GG cell (cm^3) can be defined in this section. The volume can be also set at `[region]` section by CG. If the volume is double defined, the value defined in this `[volume]` section is used. The volume is utilized in the tally sections. If you do not set volume, it has 1.0 cm^3 volume.

```
    [ V o l u m e ]
        reg              vol
          1            1.000000
         11            5.000000
( { 2 - 5 } 8 9 )      2.000000
   ( 11 12 15 )        3.000000
         16            6.000000
        ....           ........
        ....           ........
```

You can use the format `( { 2 - 5 } 8 9 )` for a group. In this case, you need to close a value by `( )` if it is not a single numeric value.

You can not use the lattice and universe style as `( 6 < 10[1 0 0] < u=3 )`. If you want to set cell volume in detail, use the volume definition in the tally section.

If you want to change the order of region number (`reg`) and volume (`vol`), you can set as "`vol reg`". You can use the skip operator `non`. Even if you use GG, you should write the symbol not `cell` but `reg` here.

4.13 [T e m p e r a t u r e] section

Free-Gas Thermal Temperature (MeV) for CG region and GG cell can be defined in this section. This section corresponds to TMP card but you can not set time definition. This value can be set in the [cell] section when you use GG, If the temperature is double defined, temperatures defined in this [temperature] sections are used. If you do not set this, the default value is 2.53×10^{-8}MeV.

```
    [ T e m p e r a t u r e ]
        reg                tmp
          1              1.0*1.e-8
         11              5.0*1.e-8
  ( { 2 - 5 } 8 9 )      2.0*1.e-8
     ( 11 12 15 )        3.0*1.e-8
         16              6.0*1.e-8
        ....             ........
        ....             ........
```

You can use the format ({ 2 - 5 } 8 9). In this case, you need to close a value by ( ) if it is not a single numeric value.

You can not use the lattice and universe style as (6 < 10[1 0 0] < u=3).

If you want to change the order of region number (reg) and temperature (tmp), you can set as "tmp reg". You can use the skip operator non. Even if you use GG, you should write the symbol not cell but reg here.

4.14 [Brems Bias] section

Energy bias of the bremsstrahlung process can be defined for each material in this section. This corresponds the BBREM card.

In the bremsstrahlung process, many low energy photons are generated. In case that you are interested in high energy photon, you can define bias for each energy in order to improve statistics for interest energy region.

```
[ Brems Bias ]
    material =  all or number of materials
       m1   m2   m3   .....
    num              bias
    {1-45}            1.0
      46              2.0
      47              3.0
      48              4.0
      49              5.0
```

First, you define the material for bias setting. If you set all, the next line is not necessary, else if you set numerical value, material numbers for the value should be listed in the next line.

Biases should be defined for the group from 1 to 49. Large number corresponds high energy. The bias is relative value.

If you want to replace the order of group number (`num`) and bias (`bias`), you can set as "`bias num`". You can use the skip operator `non`.

4.15 [Photon Weight] section

Photon production weight for CG region and GG cell can be defined in this section. This corresponds the PWT card.

When the photon production weight W_i for cell i takes positive value, photon is generated if its weight is larger than $W_i \times I_s/I_i$. Unless, the photon is treated by the Russian roulette method. In this case, I_s and I_i are the importance of neutron for source and creation point cells, respectively.

In the case the photon production weight W_i for cell i takes negative value, photon is generated if its weight takes larger than $W_i \times W_s \times I_s/I_i$. Unless, the photon is treated by the Russian roulette. In this case, W_s is a weight of neutron before nuclear reaction.

If $W_i = 0$, one photon is generated by neutron reaction. If $W_i = -1.0e6$, photon creation is ignored in the cell. By the default, $W_i = -1$.

```
    [ Photon Weight ]
        reg             pwt
          1             0.1
         11             0.3
( { 2 - 5 } 8 9 )       0.5
   ( 11 12 15 )         0.1
         16             0.9
        ....            ........
        ....            ........
```

You can use the format `( { 2 - 5 } 8 9 )`. In this case, you need to close a value by `(  )` if it is not a single numeric value.

You can not use the lattice and universe style as `( 6 < 10[1 0 0] < u=3 )`.

If you want to change the order of region number (`reg`) and weight (`pwt`), you can set as "`pwt reg`". You can use the skip operator `non`. Even if you use GG, you should write the symbol not `cell` but `reg`.

4.16 [Forced Collisions] section

The forced collisions are used for improving tally statistics or thin target analysis by enlarging the collision probabilities. When specified particle comes into a region defined as the forced collision region, the particle is divided into two particles. One has a weight by (penetration probability) × (its weight), this particle pass through to the next region. The other has a weight of (1 - penetration probability) × (its weight), and it is forced to collide with a target in the region. The collide position is decided by cross sections and random number. Regions and factors for the forced collisions can be defined in this section. Non-defined regions are set factor zero.

Maximum 6 [forced collisions] sections are allowed to be defined in a input file.

```
    [ Forced Collisions ]
        part =  proton  neutron
        reg               fcl
          1              1.000000
          11             0.500000
( { 2 - 5 } 8 9 )        0.200000
   ( 11 12 15 )          0.300000
( 6<10[1 0 0]<u=3 )     -0.500000
        ....             ........
        ....             ........
```

You set particle as `part =` in the first line. The default is `part = all`. `part =` is the same format as in tally definition.

If you want to replace the order of region number (`reg`) and (`fcl`), you can set as "`fcl reg`". You can use the skip operator `non`. Even if you use GG, you should write the symbol not `cell` but `reg` here.

You can use the format `( { 2 - 5 } 8 9 )`, and you can use the lattice and universe style as `( 6 < 10[1 0 0] < u=3 )`. But you need to close a value by `( )` if it is not a single numeric value. By using this format you can set different forced collision factor for each lattice. If the same cell is re-defined, the value, which is defined at first, is used.

The forced collision factor `fcl` means, 0: no forced collision, $|fcl| > 1$: is an error, and $|fcl| \leq 1$: multiply forced collision probability by fcl, instead the weight is reduced by $1/fcl$ times.

We have two options between the multi scattering and the weight cut off in the forced collision region. When $fcl < 0$, secondary particles produced by forced collisions are treated by the normal process. In this case, weight cut off is not performed. When $fcl > 0$, the forced collision is also applied to secondary particles. In this case weight cut off is performed. Even if a particle is killed by this weight cut off, of course the particle is tallied before killed.

There is a possibility that all particles are killed by this weight cut off if you set the weight cut off and the forced collisions without consideration. You should give attention about it.

4.17 [M a g n e t i c F i e l d] section

4.17.1 Charged particle

You can set a magnetic field in the P_{HITS} calculation. Region or cell number (reg), magnetic field type (typ), half distance of magnets (gap)(cm), magnetic field intensity (mgf), transformation (trcl) and time dependence (time) should be defined as

```
[ Magnetic  Field ]
  reg              typ    gap           mgf          trcl     time
    1               4     10.00000      -5.956540       3     non
    2               4     10.00000       6.416140       1     non
    3               2     10.00000      -7.611980       0     0.0
    4               2     10.00000       3.516000       0     pi/2
   ( 150 < 61 )     4     13.00000       7.880140       2     non
   ( 150 < 62 )     4     13.00000      -7.440800       2     non
   ( 150 < 63 )     4     13.00000       9.441010       2     non
   ( 150 < 64 )     4     13.00000      -8.295220       2     non
   ( 150 < 65 )     4     13.00000       3.694830       2     non
   ( 150 < 66 )     4     13.00000      -2.099350       2     non
   ...             ...    ........       ........      ...     ...
   ...             ...    ........       ........      ...     ...
```

The column of trcl is omissible. The zero for trcl means no transformation. The time is a parameter of user defined time dependent magnetic field. The column of time is also omissible. The "non" for time means no time dependence. Two subroutines, usrmgt1.f and usrmgt2.f are included in the source as user defined subroutines for the time dependent magnetic field. The former is for Wobbler magnet, and the latter is for pulse magnet for neutron optics. You can choose these two subroutine by usrmgt=1, 2 in the parameter section. For the Wobbler magnet, "time" means phase of the magnet, starting time for pulse magnet, respectively.

In the above expression, reg is region number, typ can take 2 or 4 for dipole electromagnet, or quadrupole electromagnet respectively. mgf denotes the strength of the magnetic field (kG), and trcl is the coordinate transformation number defined in [transform] section.

You can use the format ({ 2 - 5 } 8 9), and you can use the lattice and universe style as (6 < 10[1 0 0] < u=3). But you need to close a value by ( ) if it is not a single numeric value.

By using this format, you can set different magnetic field for each lattice. If a cell is re-defined, the value, which is defined at first, is used.

In the case of dipole magnet, the distances gap make no sense, but set any numeric. The magnetic field is available not only in the void region, but also in the material where the normal reaction can be occurred.

z-axis is assumed to be the center of the magnetic field. The direction of the magnetic field is positive direction of y-axis for dipole, i.e. the positive charge particle is bent to positive direction of x-axis when it goes to positive direction of z-axis. For quadrupole, the positive particle is converged in x-axis, diverged in y-axis when it goes to positive direction of z-axis. You need the coordinate transformation by trcl for different geometrical situation.

4.17.2 Neutron

The definition of the magnetic field for neutron is almost the same as for charged particles. Here we describe the detail of the magnetic field for neutron.

```
[ Magnetic  Field ]
  reg    typ    gap           mgf          trcl     polar     time
    1     60    0.00000      35000.0          3       non      non
    2     61    0.00000      35000.0          1         1      non
    3    106    5.00000       7130.0          0         0      non
    4    104    0.00000        3.5            0       non      5.0
    5    102    0.00000        0.20           0       non      non
    6    101    3.00000       7130.0          2         1      non
    7    103    0.00000      35000.0          0        -1      non
   ...   ...    ........      ........       ...      ...      ...
   ...   ...    ........      ........       ...      ...      ...
```

We can not take into account of the gravity nor additional dipole magnet. For `60` case, it is assumed that the spin always keeps parallel or anti-parallel to the magnet field. For `61` case, we solve the coupled equation of motion between the spin and the magnetic field. Then the spin flip can be occurred in the region with weak magnetic field. The strength of the magnetic field is specified in the unit of [T/m^2] in mgf column.

For the types above 100, we consider the coupled equations of the spin and the magnetic field. In addition, the effects of the gravity and additional dipole field can be taken into account. `106` is sextupole, `104` quadrupole, and `102` dipole, respectively. The strength of additional quadrupole magnet (z-direction) is give by the column of gap in the unit of [T].

For `101` type, the magnetic field is defined by the user program file, usrmgf1.f. In this user program, the data measured by the neutron optics group in JAERI are read from the file and used the calculation. The strength of this field is renormalized by the value of mgf.

For `101` type, the magnetic field is also defined by the user program file, usrmgf3.f. In this user program, there is a simple sextupole magnet field as same as in `106` type.

The neutron goes into the magnetic field with the initial spin if it is defined in the source section. If not, the initial spin is defined at the moment when the neutron goes into the magnetic field. The ratio of the number of parallel and anti-parallel spin to the magnetic field is determined by the polarization defined by the `polar` column. `non` in `polar` column means 0 polarization. The polarization is defined as

$$P = \frac{\phi_+ - \phi_-}{\phi_+ + \phi_-},$$

here, ϕ_+ and ϕ_- are the number of the parallel and anti-parallel particles.

4.18 [C o u n t e r] section

The counter function can be defined in this section. Three counters can be used in tally sections. The counter counts when (1) a particle comes into specified region, (2) a particle goes out specified region, (3) a particle takes scattering in specified region, and (4) a particle reflects back on a certain boundary of the region. You can set one progress value of the counter from -9999 to 9999, or zero set (10000). Counter values are attached to particles. Secondary particles produced in the collisions have the same counter value of parent. Capacity of the counter is from -9999 to 9999. Counter changes only this range. You can set the counter for each particle by using `part =` definition, and you can exclude some particles from the counter actions by `*part =` definition.

```
[ C o u n t e r ]
   counter = 1
      part = neutron proton
    reg     in     out    coll     ref
      1      1   10000       0       0
     11      1   10000       0       0
   counter = 2
     *part = proton deuteron triton 3he alpha nucleus
    reg                          in     out    coll
     ( { 2 - 5 } 8 9 )           -1       0       1
   counter = 3
      part = 208Pb
    reg                          coll
     ( 11 12 15 )                   5
     ( 6<10[1 0 0]<u=3 )          100
    ....             ........
    ....             ........
```

If you want to change the order of region number (`reg`), (`in`), (`out`), (`coll`), and (`ref`), set as "`reg coll in out ref`". You can use the skip operator `non`. At least one must be defined in the "`in out coll ref`". If nothing is defined, it is assumed no counter. Numeric gives one progress value of the counter. 10000 means zero set. The initial counter value of source particle is zero.

You can use the format `( { 2 - 5 } 8 9 )`, and you can use the lattice and universe style as `( 6 < 10[1 0 0] < u=3 )`. But you need to close a value by `(  )` if it is not a single numeric value.

In the definition of `part =` , you can specify particles up to 20 particles. For nucleus, you can use the expression like `208Pb` and `Pb`. The later case, `Pb`, denotes all isotopes of `Pb`.

4.19 [Reg Name] section

Region names and size for graphic output by gshow and 3dshow tallies are defined in this section. By default, region name is the region number.

```
[ Reg Name ]
            reg       name        size
              1       cover          1
              2        body        0.5
              3     {cell 2}          2
              4     {cell 3}          2
      { 5 - 8 }       tube           3
    ....            ........
    ....            ........
```

If you want to replace the order of region number (`reg`), (`name`), and (`size`), set as "`reg size name`". You can use the skip operator `non`. At least one must be defined in the "`name size`". If nothing is defined, it is assumed to be default. You can use the format `{ 4 - 7 }`, but the `( { 4 - 7 } 9 10 )` format can not be used. If you need to use blanks in name definition, the name must be closed by { } as the example. If you want to use `( )`, you should write `\( \)`. In the name, you cannot use { } .

4.20 [Mat Name Color] section

Material names, size and colors for graphical output by gshow and 3dshow tallies are defined in this section. By default, the name is set as material number and the color is set automatically.

```
[ Mat Name Color ]
            mat       name        size               color
              0       void           1           lightgray
              1        air         0.5          yellowgreen
              2     {mat 2}           2         orangeyellow
              3     {mat 3}           2  { 0.067 0.600 1.00 }
        { 4 - 7 }          Fe         3           mossgreen
      ....            ........
      ....            ........
```

If you want to replace the order of material number (`mat`), (`name`), (`size`), and (`color`), set as "`mat color size name`". You can use the skip operator `non`. You must define at least one parameter in "`name`" and "`color`". If no definition, the default values are used.

You can use the format `{ 4 - 7 }`, but the `( { 4 - 7 } 9 10 )` format can not be used. If you need to use blanks in name definition, the name must be closed by `{ }` as the example. If you want to use `( )`, you should write `\( \)`. In the name, you cannot use `{ }` . The color definition is based on the format in $A_N G_E L$. Set color by symbol (`r bbb yy`), name (`red orange blue`), or HSB numeric *H*(*hue*) *S* (*chroma*) *B*(*brightness*). In the case HSB numeric definition, close each numeric by `{ }`. If only one HSB numeric is defined, chroma and brightness are set 1.

Color symbols, names, and HSB numerics are shown from next page.

Table 4.50: gray scale

記号	HSB定義	出力	名前
W	–1.0		white
O	–0.8		lightgray
K	–0.6		gray
J	–0.4		darkgray
F	–0.2		matblack
E	–0.0		black

Table 4.51: Color definition by symbols

記号	HSB定義	出力	名前
R	1.000		red
RR	0.933		orange
RRR	0.867		–
Y	0.800		yellow
YY	0.733		–
YYY	0.667		–
G	0.600		green
GG	0.533		–
GGG	0.467		–
C	0.400		cyan
CC	0.333		–
CCC	0.267		–
B	0.200		blue
BB	0.133		violet
BBB	0.067		magenta

Table 4.52: Color definition by names and HSB numerics

名前	出力	HSB定義		
darkred		1.000	1.000	0.600
red		1.000	1.000	1.000
pink		1.00	0.500	1.000
pastelpink		0.900	0.500	1.000
orange		0.933	1.000	1.000
brown		0.900	1.000	0.500
darkbrown		0.900	1.000	0.300
pastelbrown		0.900	0.600	0.500
orangeyellow		0.867	1.000	1.000
camel		0.800	0.700	0.700
pastelyellow		0.800	0.700	1.000
yellow		0.800	1.000	1.000
pastelgreen		0.700	0.600	1.000
yellowgreen		0.700	1.000	1.000
green		0.600	1.000	1.000
darkgreen		0.600	1.000	0.600
mossgreen		0.500	1.000	0.300
bluegreen		0.500	1.000	1.000
pastelcyan		0.400	0.400	1.000
pastelblue		0.250	0.400	1.000
cyan		0.400	1.000	1.000
cyanblue		0.400	1.000	0.500
blue		0.200	1.000	1.000
violet		0.133	1.000	1.000
purple		0.100	1.000	0.500
magenta		0.067	1.000	1.000
winered		0.002	0.800	0.700
pastelmagenta		0.067	0.600	1.000
pastelpurple		0.100	0.400	0.500
pastelviolet		0.133	0.400	1.000

4.21 [Mat Time Change] section

By this section, you can change the material of certain cells to the other material as a function of time. This function is useful to describe a shutter of beam line, T0 chopper and the other devices for neutron optics. The unit of time is nsec.

```
[ Mat Time Change ]
              mat       time      change
                1       50.0       11
                2      100.0       12
                3     1000.0        0
              ....     ........
              ....     ........
```

In the above example, the material 1 is changed to material 11 at t=50.0 nsec, 2 to 12 at 100 nsec and 3 to void at 1000 nsec. If you want to replace the order of the initial material (mat), time (time) and the final material (change), set as "mat change time". You can use the skip operator non. These three columns are always necessary to define the mat time change function.

4.22 [Super Mirror] section

The reflection of low energy neutron by super mirror is defined by this section. We assume the following empirical formula to describe the reflectivity of the super mirrors.

$$R = \begin{cases} R_0 & \text{if } Q \le Q_c \\ \frac{1}{2}R_0\,(1-\tanh\left[(Q-mQ_c)/W\right])\,(1-\alpha(Q-Q_c)) & \text{if } Q > Q_c \end{cases}$$

where Q is the scattering vector (in Å^{-1}) defined by

$$Q = |\mathbf{k}_i - \mathbf{k}_f| = \frac{4\pi \sin\theta}{\lambda}.$$

The value of m is a parameter determined by the mirror material, the bilayer sequence and the number of bilayers. Q_c is the critical scattering wave vector for a single layer of the mirror material. At higher values of Q, the reflectivity starts falling linearly with a slope α until a cutoff at $Q = mQ_c$. The width of the cutoff is denoted W.

These parameters are defined as

```
[ Super Mirror ]
   r-in          r-out    mm   r0    qc       am    wm
   {2001-2020}    3001     3    0.99  0.0217   3.0   0.003
   2500           3500     3    0.99  0.0217   3.0   0.003
   2600           3600     3    0.99  0.0217   3.0   0.003
   ....            ....     ..   ...   ....     ...   ...
   ....            ....     ..   ...   ....     ...   ...
   ....            ....     ..   ...   ....     ...   ...
```

The reflection surface is defined by the surface between `r-in` and `r-out`. You can use the format `( { 2 - 5 } 8 9 )`, and you can use the lattice and universe style as `( 6 < 10[1 0 0] < u=3 )` in these definitions. The remaining parameters in above expression denote m by `mm`, R_0 by `r0`, Q_c by `qc` in Å^{-1}, α by `am` in Å, and W by `wm` in Å^{-1}.

We restrict this function only to neutrons for the case that its energy is less than 10 eV or sinθ is greater than 0.001, the latter is due to roughness of the surface.

4.23 [Elastic Option] section

In this section, you can set some parameters for user defined elastic option for low energy neutrons. By this function, you can change angle distributions of elastic collisions of data based neutron reactions. We prepare two sample routines, usrelst1.f and usrelst2.f. You can choose one of these two by usrelst=1, 2 in the parameter section. You should define the regions to which this function is applied and 4 parameters as,

```
[ Elastic Option ]
              reg       c1       c2       c3       c4
                1        5        1      3.3      0.4
                2        1        1      1.1      0.7
                3        3        1      0.3      0.8
              ....     ...      ...      ...      ...
              ....     ...      ...      ...      ...
```

If you want to replace the order of region number (reg), (c1 c2 c3 c4), set as "reg c3 c2 c1 c4". You can use the skip operator non. You can use the format { 4 - 7 }, but the ({ 4 - 7 } 9 10) format can not be used.

The sample routine of usrelst1.f is for Bragg scattering based on the data base, and usrelst2.f for any type of angular distribution described by an analytic formula.

4.24 [T i m e r] section

The timer function can be defined in this section. The timer controls the time of each particle when (1) a particle comes into specified region, (2) a particle goes out specified region, (3) a particle takes scattering in specified region, and (4) a particle reflects back on a certain boundary of the region. You can set the time to be zero(-1), stopped(1) or nothing(0).

```
[ T i m e r ]
    reg     in     out    coll     ref
      1      0      -1       0       0
     11      1       0       0       0
    ....   ....    ....    ....    ....
    ....   ....    ....    ....    ....
    ....   ....    ....    ....    ....
```

If you want to replace the order of region number (`reg`), (`in`), (`out`), (`coll`), and (`ref`), set as "`reg coll in out ref`". You can use the skip operator `non`. At least one must be defined in the "`in out coll ref`". If nothing is defined, it is assumed no action.

You can use the format `( { 2 - 5 } 8 9 )`, and you can use the lattice and universe style as `( 6 < 10[1 0 0] < u=3 )`. But you need to close a value by `(  )` if it is not a single numeric value.

4.25 [delta ray] section

In this section, you can set parameters used in the function to generate knocked-out electrons so-called δ-rays, which are produced along the trajectory of a charged particle in materials, as secondary particles. In the *PHITS* calculation, an energy transfer to the material is estimated by Linear Energy Transfer (LET; dE/dx), and is assumed to be deposited only on the particle trajectory. However, it is well known that owing to a high energy δ-ray the energy deposition is spread far away from the orbit of the primary particle. You can take the effect of δ-rays into account using this function. The production cross sections of δ-rays from those particles in liquid water were calculated using the model proposed by Butts and Katz,[6] considering the relativistic collision dynamics.

You can set a threshold energy $E_{\rm th}$ (MeV) for each region except outer void to control the production of δ-rays. As the secondary particle, δ-rays with energies above $E_{\rm th}$ are explicitly generated and transported. For lower energies than $E_{\rm th}$, the deposition energies from δ-rays are included in LET. A minimum energy of $E_{\rm th}$ you can set is 0.001MeV (= 1keV). It is noted that in case of very low $E_{\rm th}$ or setting of a material thiner than 10μg/cm^2, a behavior of the charged particle slightly changes. This is because the effective stopping power of the charged particles becomes smaller than its real value due to too many delta-ray productions. A default value of $E_{\rm th}$ is $1.e+10$, i.e. δ-rays are not produced in the *PHITS* calculation except for setting the $E_{\rm th}$ parameter in this section. The region number and $E_{\rm th}$ are given by `reg` and `del`, respectively. Set these parameters as follows.

```
[ delta ray ]
    reg     del
      1     0.1
     11     1.0
    ....   ....
    ....   ....
```

You can use the format `( { 2 - 5 } 8 9 )`. But you need to close a value by `( )` if it is not a single numeric value. You cannot use the lattice and universe style as `( 6 < 10[1 0 0] < u=3 )`. If you want to replace the order of region number (`reg`) and the threshold energy (`del`), set as "`del reg`". You can use the skip operator `non`. Even if you use GG, use the symbol not `cell` but `reg` here.

4.26 [Multiplier] section

In this section, you can define a multiplier set, which consists of factors depending on energies of particles, to multiply results of the [t-track] tally. When you use this set, you have to define multiplier subsections in the [t-track] section. For example, you can utilize this function for dose estimation using any dose conversion factor.

In one [multiplier] section, you can define only one multiplier set. The maximum number of the [multiplier] section defined in an input file is 100. Format of this section is as follows.

```
[ Multiplier ]
  number = -201
  interpolation = log
  ne = 10
       20.0       2.678
       30.0       7.020
       50.0       18.50
       100.0      24.26
       200.0      16.13
       500.0      10.51
       1000.0     10.55
       2000.0     10.98
       5000.0     12.10
       10000.0    12.45
```

The ID number of the set is determined by number, which must be between −299 and −200, and is used in the [t-track] section. You can choose which log-log or lin-lin as the interpolation method of the given data table, by setting interpolation = log or lin, respectively. The number of the energy point is given by ne, and data sets for the point and the factor are defined, respectively, below ne.

When you use the multiplier set defined in this section, you have to use multiplier option of the [t-track] section. The basic format is given as (C k), where C is a normalization factor and k is the ID number of the set. It is noted that k should be negative. Format of the multiplier subsection is given as follows.

```
multiplier =  number of material
      part = neutron
      emax = 1000
    mat     mset1          mset2
      1   ( 1    -201 )  ( 2 -202 )
      2   ( 1.2  -201 )  ( 3 -202 )
    ....   ....     ....   ....
    ....   ....     ....   ....
```

The line of multiplier = specifies the number of material where the multiplication is considered. You can use all instead of the number. For this case, one should also use all for mat column below. The second line of part = defines the particles considered. The maximum number of the particle is 6 and all can be also used, which is default. The multiplication affects only these considered particles. The third line of emax = defines the maximum energy of the multiplication. If emax is omitted, it is automatically defined as the maximum energy given in the [multiplier] section. The number of mat column is the material number which is considered to be multiplied. The columns of mset1, mset2 define the multiplier set. The maximum 6 multiplier can be set. For each set, the result is printed out. You can define several multiplier subsections in one [t-track] section, but you should set the number of the multiplier sets to be equal in each subsection.

5 Common parameters for tallies

PHITS has the following tally functions.

Table 5.1: Tally sections

name	explanation
[t-track]	Track length tally definition
[t-cross]	Surface crossing tally definition
[t-heat]	Heat developing tally definition
[t-deposit]	Deposit tally definition
[t-deposit2]	Deposit2 tally definition
[t-yield]	Residual nuclei yield tally definition
[t-product]	Produced particle tally definition
[t-dpa]	DPA tally definition
[t-let]	LET tally definition
[t-sed]	SED tally definition
[t-time]	Time tally definition
[t-star]	Star density tally definition
[t-dchain]	Dchain tally definition
[t-userdefined]	User defined tally definition
[t-gshow]	Region surface display definition for graphical plot
[t-rshow]	Physical quantity region display definition for graphical plot
[t-3dshow]	3D graphical geometry plot definition

Common parameters used in these tallies are described below.

5.1 Geometrical mesh

In the tallies shown by Table 5.1, CG and GG region mesh (reg), r-z scoring mesh (r-z), and xyz scoring mesh (xyz) can be used for geometrical mesh of tallying area.

You can choose one mesh from

```
mesh = [ reg, r-z, xyz ]
```

5.1.1 Region mesh

The region mesh defined by the region number or the cell number can be written by

```
mesh = reg
 reg = 1 2 3 4 5 ( 10 11 ) 50
```

each region number or cell number is separated by blank. If you want to combine some regions, use( ). The following format can be used for defining sequential region numbers.

```
mesh = reg
 reg = { 1 - 5 } ( 10 11 ) ( 6 < 10[1 0 0] < u=3 )
```

In the format {n1 - n2} (n1 is smaller than n2), you can specify regions from n1 to n2. You can't specify like (n1 - n2). Styles ({ }) and (all) can be used, but { ( ) } can not be used. You can use the lattice and universe style as (6 < 10[1 0 0] < u=3). By using above format, you can tally from each lattice individually. And if you set region as reg = all, all regions become tallying region. However, cells which do not belong to bottom level, are not included.

5.1.2 Definition of the region and volume for repeated structures and lattices

When you define regions including repeated structures and lattices, you must close your definition by ( ). A level structure is indicated by <. In the case an intermediate level has the lattice structure, you can specify lattices using [ ] represented by the lattice coordinate (s, t, u), after the cell number as 160[1:2 3:6 1:1]. In this example, lattices, which from 1 to 2 in s direction, 3 to 6 in t direction, and 1 in u direction, are defined. Or you can specify individually as 160[1 3 4, 2 3 4, 3 3 4]. The style ( ) in one level can be used to combine some regions. See next example.

List 5.1 ● mesh = reg example (1)

```
1:      mesh = reg
2:      reg = (all)
3:            ({ 201 - 205 })
4:            ( 161 < 160[1:2 3:6 1:1] )
5:            ( (201 202 203 204) < (161 162 163 ) )
6:            ( ( 90 100 ) 120 < 61 ( 62 63 ) )
```

This region mesh definition is echoed as

List 5.2 ● mesh = reg example (2)

```
1:      mesh =  reg           # mesh type is region-wise
2:       reg = ( all ) ( { 201 - 205 } ) ( 161 < 160[ 1:2 3:6 1:1 ] ) ( (
3:             { 201 - 204 } ) < ( { 161 - 163 } ) ) ( ( 90 100 ) 120 < 61
4:             ( 62 63 ) )
5:      volume                # combined, lattice or level structure
6:      non   reg      vol     # reg definition
7:        1  10001   8.1000E+01 # ( all )
8:        2  10002   5.0000E+00 # ( { 201 - 205 } )
9:        3  10003   1.0000E+00 # ( 161 < 160[ 1 3 1 ] )
10:       4  10004   1.0000E+00 # ( 161 < 160[ 2 3 1 ] )
11:       5  10005   1.0000E+00 # ( 161 < 160[ 1 4 1 ] )
12:       6  10006   1.0000E+00 # ( 161 < 160[ 2 4 1 ] )
13:       7  10007   1.0000E+00 # ( 161 < 160[ 1 5 1 ] )
14:       8  10008   1.0000E+00 # ( 161 < 160[ 2 5 1 ] )
15:       9  10009   1.0000E+00 # ( 161 < 160[ 1 6 1 ] )
16:      10  10010   1.0000E+00 # ( 161 < 160[ 2 6 1 ] )
17:      11  10011   4.0000E+00 # ( ( { 201 - 204 } ) < ( { 161 - 163 } ) )
18:      12  10012   2.0000E+00 # ( ( 90 100 ) < 61 )
19:      13  10013   1.0000E+00 # ( 120 < 61 )
20:      14  10014   2.0000E+00 # ( ( 90 100 ) < ( 62 63 ) )
21:      15  10015   1.0000E+00 # ( 120 < ( 62 63 ) )
```

In the input, it looks only 5 regions defined, but in the input echo, you can see 15 regions are defined for tally. In this input echo, region numbers are defined automatically starting from 10001, and the volume of each cell is set 1 because of no [volume] definition.

We explain the detail of 15 regions appears in the volume description of this input echo.

First for `( all )`, 81 cells are defined in the bottom level, so the volume of `( all )` is set 81. If the volume of the cell is defined correctly in the `[volume]` section, you don't need to define the volume here again.

Next for `({ 201 - 205 })`, this combined region has volume 5 in the echo, since this combined regions have 5 cells of bottom level. This is also not required to re-define here if the volume is set in the `[volume]`section.

For `( 161 < 160[1:2 3:6 1:1] )`, the region 161 is included as a lattice in region 160. In this expression in the lattice coordinate system, 8 lattices of the region 160 from 1 to 2 in *s* direction, 3 to 6 in *t* direction, and 1 in *u* direction, are used for the tally. In the echo, the number of regions in bottom level is echoed 1. In the case, you have to specify the volume by yourself by the volume definition below.

For `( (201 202 203 204) < (161 162 163 ) )`, some regions are defined in each level, but these are all closed by `( )`, so it means one region as a whole. In this case, given volume by the echo is not correct, so set volume manually by the volume definition below.

For `( ( 90 100 ) 120 < 61 ( 62 63 ) )`, there are two independent regions in each level, so 4 regions are defined here. In this case given volume by the echo is not correct too, so set volume manually in the `[volume]` section.

You can set volume as below.

```
mesh = reg
 reg = 1 2 3 4 ( 5 < 12 ) ( {13 - 17} )
volume
 reg       vol
   1       1.0000
   2       5.0000
   3       6.0000
   4       1.0000
10001      6.0000
10002      5.0000
```

In above example, region numbers from 1 to 4 are set normally as you can see, but regions `( 5 < 12 )` and `( {13 - 17} )` have numbers 10001 and 10002. These big values are given in an input echo automatically. You can see and paste this settings from the input echo.

If you want to change the order of region number (`reg`) and volume (`vol`), set as "`vol reg`". You can use the skip operator `non`.

In the input echo, numbered entry is given in `non` column. When `axis = reg`, the numbered entry is used as a value of X axis. Even if you use GG, use the symbol not `cell` but `reg` here.

When you define regions in the bottom level, set same region twice as `( 3000 < 3000[1:2 3:61:1] )`.

5.1.3 r-z mesh

When you use the r-z scoring mesh, first, offsets for x and y coordinate of the center of cylinder can be defined as

```
mesh = r-z
  x0 = 1.0
  y0 = 2.0
```

This can be omissible. Then, define r and z mesh as

```
mesh   = r-z
r-type = [1-5]
   ..........
   ..........
z-type = [1-5]
   ..........
   ..........
```

Mesh definition is described later.

5.1.4 xyz mesh

When you use the xyz scoring mesh, set x, y, and z mesh as

```
mesh   = xyz
x-type = [1-5]
   ..........
   ..........
y-type = [1-5]
   ..........
   ..........
z-type = [1-5]
   ..........
   ..........
```

Mesh definition is described later.

5.2 Energy mesh

Energy mesh begins as

```
e-type = [1-5]
   ..........
   ..........
```

”e1-type” and ”e2-type” are also used in DEPOSIT2 tally. Mesh definition is described later.

5.3 LET mesh

LET mesh begins as

```
l-type = [1-5]
   ..........
   ..........
```

Mesh definition is described later.

5.4 Time mesh

Time mesh is defined as

```
t-type = [1-5]
   ..........
   ..........
```

Mesh definition is described later.

5.5 Angle mesh

Angle mesh in cross tally is defined as

```
a-type = [1, 2, -1, -2]
   ..........
   ..........
```

If `a-type` is defined by positive number, this mesh denotes cosine mesh. If `a-type` is defined by negative number, the mesh denotes angle mesh. Mesh definition is described later.

5.6 Mesh definition

There are 8 kinds of mesh definition as `e-type, t-type, x-type, y-type, z-type, r-type, a-type` and `l-type`. The format is common for every mesh types. So only the `e-type` definition is described below. For other types, replace "e" into "t", "x", "y", ... and "a". For example, replace "`ne`" as "`nt, nx, ny, ... , na`", "`emin`" as "`tmin, xmin, ymin, ... , amin`", and so on.

5.6.1 Mesh type

You can use 5 kinds of mesh type as shown below.

Table 5.2: mesh type

mesh type	explanation
1	give number of groups and mesh points by data
2	give number of groups, minimum and maximum values. mesh is divided equally by linear scale.
3	give number of groups, minimum and maximum values. mesh is divided equally by log scale.
4	give mesh width, minimum and maximum values. mesh points are given by linear scale. Number of groups is set automatically as resulting maximum value becomes same with given value, or takes larger value with small excess as possible.
5	give minimum and maximum values and log value of mesh width mesh points are given by log scale. Number of groups is set automatically as resulting maximum value becomes same with given value, or takes larger value with small excess as possible.

It is noted that you can use only 1, 2 (-1, -2) mesh types in `a-type` definition. Each mesh type format is shown in followings.

5.6.2 e-type = 1

When you use `e-type=1`, set number of group, then numerical data as

```
e-type = 1
    ne =  number of group
          data(1) data(2) data(3) data(4)
          data(5) data(6) data(7) data(8)
          .........
          .........
          data(ne+1)
```

You can use multi lines without any symbols for line connection.

5.6.3 e-type = 2, 3

When you use `e-type = 2, 3`, set number of group, minimum value, and maximum value as

```
e-type = 2, 3
    ne =  number of group
  emin =  minimum value
  emax =  maximum value
```

5.6.4 e-type = 4

When you use `e-type=4`, set mesh width, minimum value, and maximum value as

```
e-type = 4
  edel =  width of mesh
  emin =  minimum value
  emax =  maximum value
```

5.6.5 e-type = 5

When you use `e-type= 5`, set mesh width, minimum value, and maximum value as

```
e-type = 5
  edel = log(  width of mesh  )
  emin =  minimum value
  emax =  maximum value
```

In the case, mesh width is for log scale, i.e., edel = log(M_{i+1} / M_i).

5.7 Other tally definitions

5.7.1 Particle definition

You can define particles as

```
part = proton neutron pion+ 3112 208Pb
```

or

```
part = proton
part = neutron
part = pion+
part = 3112
part = 208Pb
```

See Table 3.4 for particle identification. You can also use the kf code number.

If you define all particles as

```
part = all
```

Maximum 6 particles can be define in a tally. If you want to tally more particles, use another tally sections of the same kind of tally.

If you want to tally some particles as a group, you can use `( )` as the following. The maximum number inside the `( )` is 6.

```
part = ( proton neutron ) all pion+ 3112 208Pb
```

In this case, as the first group, the sum of proton and neutron contribution is tallied, the second is the sum of all. 5 groups of the particle are printed out in this tally.

For nucleus, you can use the expression like `208Pb` and `Pb`. The later case, `Pb`, denotes all isotopes of `Pb`.

5.7.2 axis definition

X axis value for output is described here. There are many kinds of `axis` shown as (depend on kinds of tallies or geometrical meshes),

```
eng, reg, x, y, z, r, t, xy, yz, zx, rz,
cos, the, mass, charge, chart, dchain
let, t-eng, eng-t, t-e1, e1-t, t-e2, e2-t, e12, e21
```

```
axis = eng
```

You can set multiple `axis` per one tally by

```
axis = eng  x  y
```

or,

```
axis = eng
axis = x
axis = y
```

If you define multiple axes, output results are written in different files. So you need to specify multiple output files as shown in the next subsection when multiple axes are defined.

It should be noted that you can define only one `axis` in a `[t-yield]` section from ver. 2.50. This restriction was implemented to calculate statistical uncertainties correctly. If you want to define several axes in the `[t-yield]` tally, you have to set the corresponding number of `[t-yield]` sections in a input file.

5.7.3 file definition

The format to define name of output file is,

```
file = file.001  file.002  file.003
```

As described before, when you set multiple `axis`, set output files for each axis like following example.

```
file = file.001
file = file.002
file = file.003
```

5.7.4 resfile definition

The format to specify a file name of past tally in the restart calculation is,

```
resfile = file.001
```

where the file name must be written with full pathname. Even if several `resfile` parameters are set in a tally section, only the earliest one is used. `resfile` is set to `file` by default. In this case, results of the past tally are overwritten.

5.7.5 unit definition

Set output unit as

```
unit = number
```

The unit number and its meanings are described in each tally explanation.

5.7.6 factor definition

You can set normalize factor by this format.

```
factor = number
```

This value is multiplied to output values. When you use the [t-gshow] tally, this factor defines line thickness instead.

5.7.7 output definition

Set output type as

```
output = name of output
```

Details are described in each tally explanation.

5.7.8 info definition

This option defines whether detailed information is output or not. Set 0 or 1 as

```
info = 0, 1
```

5.7.9 title definition

This option is for title as

```
title = title of the tally
```

It is omissible, and in this case, default is used.

5.7.10 ANGEL parameter definition

In order to add AnG_EL parameters in tally output, define as

```
angel = xmin(1.0) ymin(1.3e-8)
```

Defined parameter is converted to the *AnGEL* format as

```
p: xmin(1.0) ymin(1.3e-8)
```

See *AnGEL* manual for details.

5.7.11 2d-type definition

When you define 2 dimensional output as `axis = xy`, you must set this `2d-type` option as

```
2d-type =  1, 2, 3, 4, 5, 6, 7
```

These 2d-types give the format of data arrange.

- `2d-type = 1, 2, 3, 6, 7`
  Data are written by below format (the example is written by FORTRAN style).

  ```
  ( ( data(ix,iy), ix = 1, nx ), iy = ny, 1, -1 )
  ```

  10 data are written in a line. Also a header for the *AnGEL* input is inserted. The *AnGEL* header is inserted by `2d-type = 1` for contour plot, `2d-type = 2` for cluster plot, `2d-type = 3` for color plot, `2d-type = 6` for cluster and contour plot, `2d-type = 7` for color and contour plot.

- `2d-type = 4`
  Data are written by below format

  ```
  do iy = ny, 1, -1
  do ix = 1, nx
     ( x(ix), y(iy), data(ix,iy) )
  end do
  end do
  ```

  3 data of `x(ix)`, `y(iy)` and `data(ix,iy)` are written in a line.

- `2d-type = 5`
  Data are written by below format

  ```
  y/x  ( x(ix), ix = 1, nx )
    do iy = ny, 1, -1
       ( y(iy), data(ix,iy), ix = 1, nx )
    end do
  ```

  `nx + 1` data are written in a line, and total `ny + 1` lines. It is useful to use in the tabular soft like Excel.

5.7.12 gshow definition

This option can be used in all tallies without [t-gshow] and [t-rshow]. If you set gshow option with xyz mesh, xy, yz, or xz axis, and 2d-type = 1, 2, or 3, *ANGEL* can create a graphical plot with region boundary and material name, or region name, or lattice number on the two dimensional output. You can also obtain graphical plots directory from the *PHITS* calculation by the "epsout" option.

```
gshow =  0, 1, 2, 3, 4
```

In above example, 0 means no gshow option, 1 means gshow with region boundary, 2 means gshow with region boundary and material name, 3 means gshow with region boundary and region name, 4 means gshow with region boundary and lattice numbers. When you increase the resolution of the plot by `resol` parameter, the indication of region name, material name and lattice number on the graph are sometimes disturbed. In this case, you should increase the mesh points instead of `resol`.

You can see your geometry plot on a graph without transport calculation by setting `icntl = 8` in the `[parameters]` section, and this gshow option. You should check whether regions are correct, and a xyz mesh resolution is good or not, before long time calculation.

5.7.13 rshow definition

You can use `rshow` definition in all tallies except for `[t-cross]` and `[t-gshow]` tallies. This option is available with region mesh, xy, yz, zx axis. This option makes a two dimensional plot in which each region is colored with the amount of its region's output value. And region boundaries, material name, or region name numbers are also displayed. The xyz mesh definition is required after this `rshow` definition Of course this definition is essential to [t-rshow] tally.

```
rshow =  1, 2, 3
x-type = [2,4]
   ..........
   ..........
y-type = [2,4]
   ..........
   ..........
z-type = [2,4]
   ..........
   ..........
```

`rshow = 0` means no rshow option, 1 means rshow with region boundary, 2 means rshow with region boundary and material name, 3 means rshow with region boundary and region name numbers. If `rshow =0`, xyz mesh definition is not required, comment out it. When you increase the resolution of the plot by `resol` parameter, the indication of region name, material name and lattice number on the graph are sometimes disturbed. In this case, you should increase the mesh points instead of `resol`.

If you use the rshow option with "reg" mesh, there is no output for the values of each region. In this case, you can not re-plot the figure because of no original data. When this `rshow` option is used, usually `axis` is set as xy, yz, and zx. But you should use in addition `axis = reg` in order to save results into another file, for re-plotting. You can re-plot figures from saved data and [t-rshow] tally function.

You can execute this option without transport calculation by using `icntl =10` in the `[parameters]` section. For `icntl =10`, *PHITS* makes a two dimensional plot for the tallies with reg mesh, xy, yz, zx axis and rshow = 1, 2, 3. In the figure, different colors are used for different materials. You should check whether regions are correct and a xyz mesh resolution is good or not, before long time calculation.

5.7.14 x-txt, y-txt, z-txt definition

If you want to change x, y, and z axis titles in the output figure, use these option. These title can not be defined in the *ANGEL* parameter.

```
x-txt = x axis title
y-txt = y axis title
z-txt = z axis title
```

5.7.15 volmat definition

The volmat parameter corrects a volume where xyz mesh crosses region boundaries. This option is effective in the case that mesh is xyz, and the material parameter is defined. This corrected volume is calculated by the Monte Carlo method for specified material. `volmat` denotes the number of scanning parallel to x, y, and z axis respectively for the Monte Carlo calculation. So If you set too large volmat, the calculation takes long time. You need to take care of it. If volmat is given by negative value, all xyz mesh is scanned. If positive value, the scanning is not performed when 8 apexes of the mesh are included in the same material.

5.7.16 epsout definition

If you set `epsout =1`, output file is treated by *ANGEL* automatically and an eps file is created. This eps file name is named by replacing the extension into ".eps". With `itall = 1` setting, the eps file is created after every batch calculation. You can monitor the *PHITS* results in real time, by displaying the eps file with the ghostview and by setting refresh function for a file updating by typing "`w`" key on the ghostview.

5.7.17 counter definition

You can make a gate to the tallying quantities by using the counter defined by `[counter]` section. Set minimum `ctmin(i)` and maximum value `ctmax(i)` for each counter. The "`i`" is the counter number from 1 to 3. By default, `ctmin(i) = -9999`, and `ctmax(i) = 9999`. When multiple counters are specified, the common part of these terms are tallied.

5.7.18 resolution and line thickness definitions

You can increase the resolution of the region boundaries in the gshow, rshow, and 3dshow with keeping xyz mesh by `resol`. Default value is 1, it is same as xyz mesh resolution. If you set `resol = 2`, the resolution becomes 2 times for each side. It is useful to draw smooth line for xyz mesh. Also you can obtain clear graphics by set `resol` larger for the 3dshow. Even if you set `resol` larger, memory usage is not changed.

The `width` shows the line thickness for gshow, rshow, and 3dshow. Default value is 0.5.

5.7.19 trcl coordinate transformation

By this `trcl` option, you can transform the coordinate of the r-z, and xyz mesh. There are two ways to define the transformation as below.

```
trcl = number
trcl = O1  O2  O3  B1  B2  B3  B4  B5  B6  B7  B8  B9  M
```

The first definition is to specify the transformation number defined in [transform] section. The next one is to define the transformation directly here with 13 parameters as same as in [transform] section. If the data are not written in a line, you can write them in multiple lines without the line sequential mark. But you need to put more than 11 blanks before data on the top of the sequential lines.

In the 3dshow tally, trcl can be used to transform the box. This will be explained in the [t-3dshow] tally section.

5.7.20 dump definition

In the [t-cross], [t-time], [t-product] tallies, information on the particles can be dumped on the file.

By the parameter of "dump =", the number of the dump data in one record is specified. If this number is given by positive number, the data are read as binary data. If negative, the data are read as ascii data. In next line, the data sequence of one record is described. The relation between the physical quantities and id number is the followings,

Table 5.3: id number of dump data (1)

physical quantities	kf	x	y	z	u	v	w	e	wt	time	c1	c2	c3	sx	sy	sz
id number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Table 5.4: id number of dump data (2)

physical quantities	name	nocas	nobch	no
id number	17	18	19	20

Here kf means the kf-code of the particles (see Table 3.4), x, y, z are coordinates (cm), u, v, w denote the unit vectors of the direction of the particle, e is the energy (MeV, or MeV/nucleon for nucleus), wt is the weight, time is the initial time (ns), c1, c2, c3 are the values of counters, and sx, sy, sz are the unit vectors of the direction of spin, respectively. name is a collision number of the particle, nocas is a current history number of this batch, nobch is a current batch number, no is a cascade id in this history. These are assumed as real*8 for the binary data, *n*(1p1e24.15) data format for the ascii data.

For an example, one record has 9 data as

```
kf  e  wt  x  y  z  u  v  w
```

To read this data, we write the parameters as

```
dump = 9
  1  8  9  2  3  4  5  6  7
```

The dumped data are written on the file define by file definition. When you use this dump parameter, axis and file are restricted to one axis and one file, and unit is always 1. The normal output of the tally is written on the file which name is "filename" with .cfg. From this file, you can get the information on the total normalization factor. In the parallel computing, files to the number of (PE−1) corresponding to each PE (Processor Element) are created for writing and reading dumped data. If you set idpara=0 or 1, a file is made in the directory named by /wk/uname/ on each of the nodes. If you set idpara=1 or 3, the each IP number is put at the end of the filename. The each PE writes down its result on only the corresponding file, and reads it from the same file in the re-calculation.

6 Tally input format

6.1 [T - T r a c k] section

Using the [T-Track] tally, you can obtain the flux in any specified region. In this tally, track length is evaluated whenever particles pass through the specified region as shown in Fig. 6.1, and the sum of the track lengths in the unit of (cm) is scored. Then, particle flux in the unit of (/cm^2/source) is determined from the scored track lengths divided by the volume of the region and the number of the source particles.

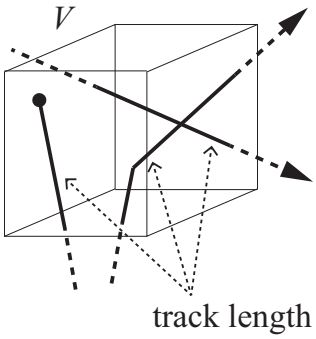


Figure 6.1: [T-Track] tally: track length (solid line) is calculated.

For an example, you can get information on the detector response in the specified region by utilizing this tally. Multiplying the flux by a cross section (in the unit of cm^2) of the detector, you can estimate the number of counts in the response.

Table 6.1: [t-track] parameter (1)

name	value	explanation
mesh =	reg, r-z, xyz	geometry mesh you need geometry mesh subsection below this option
part =	all (default), particle name	maximum 6 particles in a [t-track]
material =	(omissible) all, number of materials	You can specify materials for scoring. all : default (same as no definition) When you set number of materials, define these material numbers in the next line. You can set number of materials by negative. In the case, specified materials are not included for scoring.
(next line)	2 5 8	material numbers
e-type =	1, 2, 3, 4, 5	energy mesh You need energy mesh subsection below this option
t-type =	1, 2, 3, 4, 5 (omissible)	time mesh You need time mesh subsection below this option
unit =	1, 2, 3, 4 11, 12, 13, 14	1: [1/cm^2/source] 2: [1/cm^2/MeV/source] 3: [1/cm^2/Lethargy/source] 4: [cm/source] 11: [1/cm^2/nsec/source] 12: [1/cm^2/MeV/nsec/source] 13: [1/cm^2/Lethargy/nsec/source] 14: [cm/nsec/source]
axis =	eng, reg, x, y, z, r, xy, yz, xz, rz t	x axis value of output data 2 dimensional time axis
file =	file name	Define file names as same number of axis
resfile =	(omissible, D=file)	Define a file name of the past tally in the restart calculation. Even if several axis parameters were defined, you should specify only one resfile.
multiplier =	number of material (omissible)	multiplier for each material You need multiplier subsection below this option
factor =	(omissible, D=1.0)	normalization factor
title =	(omissible)	title
angel =	(omissible)	angel parameters
2d-type =	1, 2, 3, 4, 5, 6, 7 (omissible)	options for 2 dimensional plot

"Lethargy " in unit = 3 or 13 is a natural logarithmic unit of energy, and defined by $\ln(E_{\rm ref}/E)$ using a reference energy $E_{\rm ref}$ and a particle's energy E. If you set these units, you can obtain results per Lethargy, which are calculated by Lethargy widths, $\ln(E_{\rm high}/E_{\rm low})$, at each energy bins given in the energy mesh subsection. Here, $E_{\rm high}$ and $E_{\rm low}$ are maximum and minimum values of the energy bins, respectively.

If you set unit = 1, 2, 3, 11, 12 or 13, you obtain the mean particle flux in the specified region, which is calculated from the sum of the track lengths per source divided by the volume of the region. Noted that for reg mesh you have to set the volume in the [Volume] or [Region] section. If you do not, you obtain the particle flux for volume = 1cm^3, i.e. the sum of the track lengths per source. For r-z and xyz mesh, the volume is automatically calculated. If you set unit=4 or 14, you obtain the sum of the track length per source.

Table 6.2: [t-track] parameter (2)

name	value	explanation
x-txt = y-txt = z-txt =	(omissible) (omissible) (omissible)	x axis title y axis title z axis title
gshow =	0 (default), 1, 2, 3, 4	When mesh=xyz, axis=xy,yz,xz, region border (1), material name (2), region name (3), and LAT number(4) are plotted by the option.
rshow =	0 (default), 1, 2, 3	When mesh=reg, axis=xy,yz,xz, region border (1), material name (2), and region name (3) are plotted by the option. You need xyz mesh section below this option.
resol = width =	1 (default) 0.5 (default)	The option multiplies region line resolution by resol times with gshow or rshow option. The option defines the line thickness for gshow or rshow option.
volume reg vol	(omissible)	The option defines volume for each region for reg mesh. You need volume definitions below this option. Default values are given in input echo in the case of no definition. volume definition. See 5.1.2
iechrl =	72 (default)	Number of maximum column for volume input echo
volmat =	(omissible, D=9)	The option corrects a volume value for each mesh when material is defined by xyz mesh. (0 means no correction) Value of volmat means the number of scans for one side of xyz mesh
epsout =	0 (default), 1	If epsout is set to 1, results are plotted into eps files. This eps file name is named by replacing the extension into ".eps".
ctmin(i) = ctmax(i) =	(omissible, D=-9999) (omissible, D= 9999)	minimum value for i-th counter maximum value for i-th counter
trcl =	(omissible)	coordinate transformation number or definition for r-z or xyz mesh
gslat =	1(default), 0	1: show lattice boundary in gshow, 0: no

Using `multiplier` option in this tally, you can multiply results of the `[t-track]` tally by factors depending on energies of particles. If you define a data set in `[multiplier]` section, any factors can be used. The basic format is given as (C k), where C is a normalization factor and k is the ID number of the set. It is noted that k should be negative. Format of the `multiplier` subsection is given as follows.

```
multiplier =  number of material
      part = neutron
      emax = 1000
    mat     mset1          mset2
      1   ( 1    -201 )  ( 2 -202 )
      2   ( 1.2  -201 )  ( 3 -202 )
    ....   ....     ....   ....
    ....   ....     ....   ....
```

The line of `multiplier` = specifies the number of material where the multiplication is considered. You can use `all` instead of the number. For this case, one should also use `all` for `mat` column below. The second line of `part` = defines the particles considered. The maximum number of the particle is 6 and `all` can be also used, which is default. The multiplication affects only these considered particles. The third line of `emax` = defines the maximum energy of the multiplication. If `emax` is omitted, it is automatically defined as the maximum energy given in the `[multiplier]` section, or `dmax(i), i = 1, 2,` or `14` when you use nuclear data library. The number of `mat` column is the material number which is considered to be multiplied. The columns of `mset1`, `mset2` define the multiplier set. The maximum 6 multiplier can be set. For each set, the result is printed out. You can define several `multiplier` subsections in one `[t-track]` section, but you should set the number of the multiplier sets to be equal in each subsection.

Some parameter sets built-in *PHITS* can be used. If you set $k = -1$, a value of 1/weight is used as the multiplication factor. For $k = -2$, a value of 1/velocity is used. With $k = -101, -102, -112$, or -114, the conversion factor of proton, neutron, electron, or photon, respectively, is used. These conversion factors were estimated with a condition of Antero-Posterior geometry (AP) irradiation.[7] The unit of the dose conversion factors is (μSv/h)/(n/sec/cm^2). It should be noted that the interpolation method of conversion factor has been changed in *PHITS* ver. 2.00 from linear-linear to log-log.

You can also use the following format like FM card of MCNP.

```
multiplier =  number of material
      part = proton
      emax = 150
    mat             mset1    mset2
      1   ( 0.1236 1 1 -4 )  ( 0.0 )
      2   ( 0.0060 2 1 -4 )  ( 0.0 )
      3   ( 0.0032 3 1 -4 )  ( 0.0 )
    ....   ....     ....   ....
    ....   ....     ....   ....
multiplier =  number of material
      part = neutron
      emax = 150
    mat                    mset1                mset2
      1   ( 0.1236 1 1 -4 : -6 -8 )  ( 1.0 -1 33 0.543 )
      2   ( 0.0060 2 1 -4 : -6 -8 )  ( 1.0 -1 34 0.321 )
      3   ( 0.0032 3 1 -4 : -6 -8 )  ( 1.0 -1 35 0.678 )
    ....   ....     ....   ....
    ....   ....     ....   ....
```

In above example, the `mset1` is for heat and the `mset2` is zero for proton, attenuator set for neutron.

6.2 [T - C r o s s] section

Using the `[T-Cross]` tally, you can obtain the current or flux on any specified surface. In this tally, whenever a particle crosses the surface, current is simply added by 1, while flux is added by $1/\cos\theta$, where θ is the angle between the direction of the particle trajectory and the normal vector to the surface. In PHITS , the current and flux each other are similar but different physical quantity. The difference is due to the surface element, which is used to calculate the number of the crossing particle per unit area. The current is evaluated with division by the area of the surface S shown in Fig. 6.2. On the other hand, the flux is done with division by $S\cos\theta$. The value of S is given in the geometry mesh subsection as `area` for `reg` mesh. The S is calculated automatically for `r-z` and `xyz` mesh.

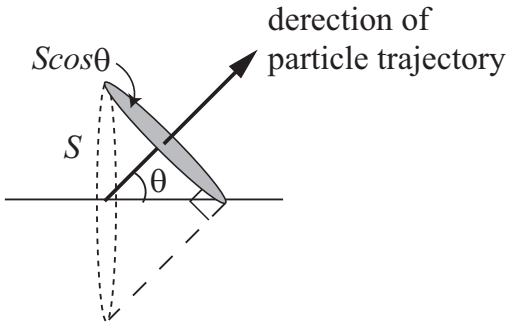


Figure 6.2: Relation between the two areas S and $S\cos\theta$.

Since the flux in this tally is evaluated with weight of $1/\cos\theta$, the result is equivalent to that obtained from the `[t-track]` tally for an extremely thin region. Consequently, you can obtain information on the detector response in the specified surface by utilizing the `[t-cross]` tally. Multiplying the flux by a cross section (in the unit of cm^2) of the detector, you can estimate the number of counts in the response.

Table 6.3: `[t-cross]` parameter (1)

name	value	explanation
`mesh =`	`reg, r-z, xyz`	geometry mesh you need geometry mesh subsection below this option
`part =`	`all` (default), particle name	maximum 6 particles in a [t-cross]
`e-type =`	`1, 2, 3, 4, 5`	energy mesh You need energy mesh subsection below this option
`a-type =`	`1, 2, -1, -2` The option is required for a-curr and oa-curr	angle mesh (1, 2 :cos, -1, -2 :degree) You need angle mesh subsection below this option
`t-type =`	`1, 2, 3, 4, 5` (omissible)	time mesh You need time mesh subsection below this option

`unit =`	`1, 2, 3, 4, 5, 6`
	
	`11, 12, 13, 14, 15, 16`
		1: [1/cm^2/source]
		2: [1/cm^2/MeV/source]
		3: [1/cm^2/Lethargy/source]
		4: [1/cm^2/sr/source]
		5: [1/cm^2/MeV/sr/source]
		6: [1/cm^2/Lethargy/sr/source]
		11: [1/cm^2/nsec/source]
		12: [1/cm^2/MeV/nsec/source]
		13: [1/cm^2/Lethargy/nsec/source]
		14: [1/cm^2/sr/nsec/source]
		15: [1/cm^2/MeV/sr/nsec/source]
		16: [1/cm^2/Lethargy/sr/nsec/source]

`axis =`	`eng, reg, x, y, z, r`
	`cos, the, t`
	`xy`
		x axis value of output data
		angle (cos, the) and time (t) mesh
		2 dimensional

You can obtain current for specified angles using the angle mesh shown in Fig. 6.3. In the cases of `unit=4, 5, 6, 14, 15, or 16`, the output is given as a quantity per unit steradian (sr) calculated by using the mesh size of the angle-bin defined in the angle mesh subsection.

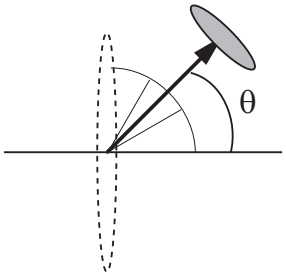


Figure 6.3: Schematic image of the tally using the angle mesh.

"Lethargy " in `unit = 3, 6, 13 or 16` is a natural logarithmic unit of energy, and defined by $\ln(E_{\mathrm{ref}}/E)$ using a reference energy E_{ref} and a particle's energy E. If you set these units, you can obtain results per Lethargy, which are calculated by Lethargy widths, $\ln(E_{\mathrm{high}}/E_{\mathrm{low}})$, at each energy bins given in the energy mesh subsection. Here, E_{high} and E_{low} are maximum and minimum values of the energy bins, respectively.

Table 6.4: [t-cross] parameter (2)

name	value	explanation
file =	file name	Define file names as same number of axis
resfile =	(omissible, D=file)	Define a file name of the past tally in the restart calculation. Even if several axis parameters were defined, you should specify only one resfile.
factor =	(omissible, D=1.0)	normalization factor
title =	(omissible)	title
angel =	(omissible)	angel parameters
2d-type =	1, 2, 3, 4, 5, 6, 7 (omissible)	options for 2 dimensional plot
output =	flux current o-curr a-curr oa-curr	flux by surface crossing current by surface crossing omni current by surface crossing omni means the energy integrated angle mesh current by surface crossing angle mesh omni current by surface crossing
x-txt = y-txt = z-txt =	(omissible) (omissible) (omissible)	x axis title y axis title z axis title
gshow =	0 (default), 1, 2, 3, 4	When mesh=xyz, axis=xy,yz,xz, region border (1), material name (2), region name (3), and LAT number(4) are plotted by the option.
resol = width =	1 (default) 0.5 (default)	The option multiplies region line resolution by resol times with gshow option. The option defines the line thickness for gshow option.
epsout =	0 (default), 1	If epsout is set to 1, results are plotted into eps files. This eps file name is named by replacing the extension into ".eps".
ctmin(i) = ctmax(i) =	(omissible, D=-9999) (omissible, D= 9999)	minimum value for i-th counter maximum value for i-th counter
trcl =	(omissible)	coordinate transformation number or definition for r-z or xyz mesh
dump = (next line)	number of data (omissible) data sequence	For mesh=reg, the information is dumped on the file. If dump is negative, data is written by ascii, if positive, by binary. define the data sequence.
gslat =	1(default), 0	1: show lattice boundary in gshow, 0: no

In the [t-cross] tally, you can use the dump option only with reg mesh, and only with reg axis. If the dump option is set, the meshes of e-type, a-type and t-type have only the meaning of the maximum and minimum values. The output option is set to be current, a-curr or oa-curr. The file in which the dump data are written is the file defined by "file = ". When you use this dump parameter, axis and file are restricted to one axis and one file. The normal output of the tally is written on the file which name is "filename" with .cfg. From this file, you can get the information on the total normalization factor. To do so, you had better set one mesh for e-type, a-type and t-type.

If you set `mesh = reg` for geometry mesh in this section, you must define crossing surface by region number for in- and out- region as an example below.

```
mesh = reg
 reg =  number of crossing surfaces
  r-in    r-out    area
    2       8      10.0
    3       8       5.0
  ( 4 5 ) ( 4 5 )   2.0
  (13<5)  (14<5)    7.0
  (13<6)  (14<6)    7.0
  (13<7)  (14<7)    7.0
   ...     ...     ....
   ...     ...     ....
```

The default order for this definition is "`r-in r-out area`". If you want to change the order, define as "`r-in r-out area`". You can use the skip operator `non`. You can use the format `( { 2 - 5 } 8 9 )`, and you can use the lattice and universe style as `( 6 < 10[1 0 0] < u=3 )`. But you need to close a value by `( )` if it is not a single numeric value. If you set `output = flux`, obtained flux is one way from r-in to r-out. If you want to set both way flux, set as the third line of above definition.

If you set `mesh = r-z`, two kinds of crossing surface are defined. One is the number of "`nz`+1" crossing surfaces for z defined by $r_i - r_{i+1}$. The other is the number of "`nr`+1" crossing surfaces for r defined by $z_i - z_{i+1}$. If r-surface coincides with the surface of outer void, the flux on this surface is not tallied.

If you set `mesh=xyz`, the number of "`nz`+1" crossing surfaces for z are defined by $x_i - x_{i+1}$ and $y_j - y_{j+1}$. In this case, x and y crossing surfaces are not defined. When you set `mesh = rz, xyz`, crossing particles are detected in both ways at defined surface. The forward definitions are, positive direction in z surface, and from center to outside in r surface.

6.3 [T - H e a t] section

[T - H e a t] gives deposit energy for optional region. Deposit energy by low energy neutron, photon, and electron can be also tallied in this tally. The heat from neutrons is usually obtained from Kerma factors with nuclear data. For `e-mode=1`, the heat from neutrons is zero, but the heat is calculated from energy loss of all charged particles and nuclei. The heat from photons is usually obtained also from Kerma factors with nuclear data. For `electron=1` with electron transport, we do not use the Kerma factors of photon, but obtain the heat from the energy loss of electrons.

Table 6.5: `[t-heat]` parameter (1)

name	value	explanation
`mesh =`	`reg, r-z, xyz`	geometry mesh you need geometry mesh subsection below this option
`axis =`	`reg, x, y, z, r,` `xy, yz, xz, rz`	x axis value of output data 2 dimensional
`file =`	`file name`	Define file names as same number of axis
resfile =	(omissible, `D=file`)	Define a file name of the past tally in the restart calculation. Even if several `axis` parameters were defined, you should specify only one `resfile`.
`material =`	(omissible) `all`, number of materials	You can specify materials for scoring. all : default (same as no definition) When you set number of materials, define these material numbers in the next line. You can set number of materials by negative. In the case, specified materials are not included for scoring.
(next line)	`2 5 8`	material numbers
`output =`	`heat` `simple`	total: Total deposit energy. total: Total deposit energy. ncut, gcut, pcut: Deposit energies of neutrons, photons, and protons below cut off energy when you set `incut>0, igcut>0, ipcut>0`, respectively. leakage: Kinetic energy of particles going out to the outer region. And, total: Total deposit energy. recoil: Kinetic energy of residual nuclei when you set cut off energy `emin(15-19)`. ionization: Deposit energy by energy-loss of charged particles. low neutron: Deposit energy calculated by neutron Kerma factors. photon: Deposit energy by Kerma factors. (If `electron=1`, contributions of electrons below cut off energy.) others: Excitation energy of residual nuclei. When you set `igamma=1`, this value comes to be 0 owing to photon emission.

Neutrons, photons, and protons below cut off energy, are not tallied in the `ncut`, `gcut`, and `pcut` component, but in the stopped particle if `incut=0`, `igcut=0`, and `ipcut=0` in the parameter section. When `incut>0`, `igcut>0`, and `ipcut>0`, they are tallied in the `ncut`, `gcut`, `pcut` part.

Table 6.6: `[t-heat]` parameter (2)

name	value	explanation
`output =` (continued)	`all`	In addition to above, Contributions of $d, t, ^3$He, α, and residual nuclei to recoil. Contributions of p, π^+, π^-, and the others to ionization. (Contributions of particles specified by `part` are output. However, they are not plotted in eps file.) stopped particle: Kinetic energy of stopped proton, neutron, π^+, π^-, and the other particles in materials. (Contributions of particles specified by `part` are output. However, they are not plotted in eps file.) others: Remaining excitation energy and fission component. When `axis` is 2 dimensional, `all` is the same as `simple`. Only total, recoil, ionization, low neutron, electron, and others are output.
`part =`	particle name (omissible)	You can specify particles. ionization and stopped particle are used as output
`unit =`	`1, 2`	1: [MeV/cm^3/source] 2: [MeV/source]
`factor =`	(omissible, D=1.0)	normalization factor
`title =`	(omissible)	title
`angel =`	(omissible)	angel parameters
`2d-type =`	`1, 2, 3, 4, 5, 6, 7` (omissible)	options for 2 dimensional plot

`x-txt =`
`y-txt =`
`z-txt =`
	(omissible)
	(omissible)
	(omissible)
		x axis title
		y axis title
		z axis title

`gshow =`	`0` (default), `1, 2, 3, 4`	When mesh=xyz, axis=xy,yz,xz, region border (1), material name (2), region name (3), and LAT number(4) are plotted by the option.
`rshow =`	`0` (default), `1, 2, 3`	When mesh=reg, axis=xy,yz,xz, region border (1), material name (2), and region name (3) are plotted by the option. You need xyz mesh section below this option.

`resol =`
 	
`width =`
	`1` (default)
	
	`0.5` (default)
		The option multiplies region line resolution
		by `resol` times with gshow or rshow option.
		The option defines the line thickness
		for gshow or rshow option.

Generally speaking, heat is an energy of ionization of charged particles. However, in the transport simulation, cutoff energy of the particle is set and the transport is stopped below the energy. Then there exist some components of heat, i.e. recoil, stopped particle, and others, in the output of the heat tally. These components may change as the parameters of the transport are changed.

Table 6.7: `[t-heat]` parameter (3)

name	value	explanation

`volume`
 	
`reg vol`
	(omissible)	The option defines volume for each region
		for reg mesh. You need volume definitions
		below this option.
		Default values are given in input
		echo in the case of no definition.
		volume definition. See 5.1.2

`iechrl =`	`72` (default)	Number of maximum column for volume input echo
`volmat =`	(omissible, D=9)	The option corrects a volume value for each mesh when material is defined by xyz mesh. (0 means no correction) Value of volmat means the number of scans for one side of xyz mesh
`epsout =`	`0` (default), `1`	If epsout is set to 1, results are plotted into eps files. This eps file is named by replacing the extension into ".eps".
`electron =`	`0` (default), `1`	electron contribution options 0: using photon KERMA factors (electron and positron should NOT be transported, otherwise their deposition energies are double counted) 1: calculating by ionization loss (electron and positron transports are required)
`ctmin(i) =` `ctmax(i) =`	(omissible, D=-9999) (omissible, D= 9999)	minimum value for i-th counter maximum value for i-th counter
`trcl =`	(omissible)	coordinate transformation number or definition for r-z or xyz mesh
`gslat =`	1(default), 0	1: show lattice boundary in gshow, 0: no

6.4 [T - Deposit] section

This tally is very similar to [t-heat] and scores dose and deposit energy distributions. The difference from [t-heat] is that this tally only counts an energy loss of charged particles and nuclei. Thus, you must use the event generator mode (`e-mode = 1`) if you would like to transport low-energy neutrons. In this tally, you can multiply any factor as a function of LET(dE/dx) in a certain material to the dose or deposit energy. This function is realized by user defined subroutine usrdfn1.f and usrdn2.f. As examples, the default program of usrdfn1.f returns the dose equivalent calculated from deposit energy multiplied with the $Q(L)$ relationship defined in the ICRP60, while that of usrdfn2.f simply does the energy loss without multiplying any factor. You can change and add any factor in this routine. In addition, using the time mesh with [timer] section, you can simulate a TOF (time of flight) detector and plot 2-dimensional graph of the correlation between the deposit energy and the TOF.

Table 6.8: `[t-deposit]` parameters(1)

name	value	explanation
`mesh =`	`reg, r-z, xyz`	geometry mesh you need geometry mesh subsection below this option
`part =`	`all` (default), particle name	maximum 6 particles in a [t-deposit]
`material =`	(omissible) `all`, number of materials	You can specify materials for scoring. all : default (same as no definition) When you set number of materials, define these material numbers in the next line. You can set number of materials by negative. In the case, specified materials are not included for scoring.
(next line)	`2 5 8`	material numbers
letmat =	(omissible)	material id for LET(dE/dx). If omitted, real material is assumed.
dedxfnc =	(omissible, D=0)	0: without, 1: use usrdfn1.f, 2: use usrdfn2.f As examples, the default program of usrdfn1.f returns the dose equivalent calculated from deposit energy multiplied with the $Q(L)$ relationship defined in the ICRP60, while that of usrdfn2.f simply does the energy loss without multiplying any factor.
`e-type =`	`1, 2, 3, 4, 5`	energy mesh You need energy mesh subsection below this option.
`t-type =`	`1, 2, 3, 4, 5` (omissible)	time mesh You need time mesh subsection below this option.
output =	`dose` `deposit`	score the energy loss of charged particles and nuclei score deposit energy distribution You need `e-type` subsection.
unit =	`1, 2, 3, 4`	1: Dose [MeV/cm^3/source] 2: Dose [MeV/source] 3: Number [1/source] ; only for `output=deposit` 4: Number [1/nsec/source] ; only for `output=deposit`

`axis =`	`eng, reg, x, y, z, r, t`
	`xy, yz, xz, rz`
	`t-eng, eng-t`
		x axis value of output data
		
		2 dimensional

`file =`	`file name`	Define file names as same number of axis
resfile =	(omissible, `D=file`)	Define a file name of the past tally in the restart calculation. Even if several `axis` parameters were defined, you should specify only one `resfile`.

Table 6.9: [t-deposit] parameter (2)

name	value	explanation
factor =	(omissible, D=1.0)	normalization factor
title =	(omissible)	title
angel =	(omissible)	angel parameters
2d-type =	1, 2, 3, 4, 5, 6, 7 (omissible)	options for 2 dimensional plot
x-txt = y-txt = z-txt =	(omissible) (omissible) (omissible)	x axis title y axis title z axis title
gshow =	0 (default), 1, 2, 3, 4	When mesh=xyz, axis=xy,yz,xz, region border (1), material name (2), region name (3), and LAT number(4) are plotted by the option.
rshow =	0 (default), 1, 2, 3	When mesh=reg, axis=xy,yz,xz, region border (1), material name (2), and region name (3) and LAT number(4) are plotted by the option. You need xyz mesh section below this option.
resol = width =	1 (default) 0.5 (default)	The option multiplies region line resolution by resol times with gshow or rshow option. The option defines the line thickness for gshow or rshow option.
volume reg vol	(omissible)	The option defines volume for each region for reg mesh. You need volume definitions below this option. Default values are given in input echo in the case of no definition. volume definition. See 5.1.2
iechrl =	72 (default)	Number of maximum column for volume input echo
volmat =	(omissible, D=9)	The option corrects a volume value for each mesh when material is defined by xyz mesh. (0 means no correction) Value of volmat means the number of scans for one side of xyz mesh
epsout =	0 (default), 1	If epsout is set to 1, results are plotted into eps files. This eps file is named by replacing the extension into ".eps".
ctmin(i) = ctmax(i) =	(omissible, D=-9999) (omissible, D= 9999)	minimum value for i-th counter maximum value for i-th counter
trcl =	(omissible)	coordinate transformation number or definition for r-z or xyz mesh
gslat =	1(default), 0	1: show lattice boundary in gshow, 0: no

In this tally, one can only score the energy loss of charged particles. So you cannot get the sum of the energy loss for a specific particle which goes into the tally region by using part = in this tally section. In order to tally the energy loss for each projectile particle going into the tally region, you should define the counter with part = in [counter] section and ctmin, ctmax in this tally section.

6.5 [T - Deposit2] section

This tally scores deposit energy distribution in two regions and plot the correlation between two deposit energies. By this, one can simulate, for an example, dE, E counters and plot the correlations in 2-dimensional graph. In this tally, as in the [t-deposit] tally, you can multiply any factor as a function of LET(dE/dx) in a certain material to the dose or deposit energy. This function is realized by user defined subroutine usrdfn1.f and usrdn2.f. As examples, the default program of usrdfn1.f returns the dose equivalent calculated from deposit energy multiplied with the $Q(L)$ relationship defined in the ICRP60, while that of usrdfn2.f simply does the energy loss without multiplying any factor. You can change and add any factor in this routine. In addition, using the time mesh with [timer] section, you can simulate a TOF (time of flight) detector and plot 2-dimensional graph of the correlation between the deposit energy and the TOF.

Table 6.10: `[t-deposit2]` parameters(1)

name	value	explanation
`mesh =` `reg =`	`reg` `r1 r2`	geometry mesh, only `reg` Two region numbers should be written.
`part =`	`all` (default), particle name	maximum 6 particles in a [t-deposit2]

`letmat1 =`
 	
`letmat2 =`
	(omissible)
	
	(omissible)
		material id for LET(dE/dx) of region r1
		If omitted, real material is assumed.
		material id for LET(dE/dx) of region r2
		If omitted, real material is assumed.

dedxfnc1 = dedxfnc2 =	(omissible, D=0) (omissible, D=0)	for region r1, 0: without, 1: use usrdfn1.f, 2: use usrdfn2.f for region r2, 0: without, 1: use usrdfn1.f, 2: use usrdfn2.f As examples, the default program of usrdfn1.f returns the dose equivalent calculated from deposit energy multiplied with the $Q(L)$ relationship defined in the ICRP60, while that of usrdfn2.f simply does the energy loss without multiplying any factor.

`e1-type =`
 	
`e2-type =`
	`1, 2, 3, 4, 5`
	
	`1, 2, 3, 4, 5`
		energy mesh for region r1
		You need energy mesh subsection below this option
		energy mesh for region r2
		You need energy mesh subsection below this option

`t-type =`	`1, 2, 3, 4, 5` (omissible)	time mesh You need time mesh subsection below this option
unit =	`1, 2`	1: Number [1/source] 2: Number [1/nsec/source]

`axis =`	`eng1, eng2, t,`
	`e12, e21, t-e1, t-e2`
	`e1-t, e2-t`
		x axis value of output data
		2 dimensional

`file =`	`file name`	Define file names as same number of axis
resfile =	(omissible, `D=file`)	Define a file name of the past tally in the restart calculation. Even if several `axis` parameters were defined, you should specify only one `resfile`.

Table 6.11: [t-deposit2] parameter (2)

name	value	explanation
factor =	(omissible, D=1.0)	normalization factor
title =	(omissible)	title
angel =	(omissible)	angel parameters
2d-type =	1, 2, 3, 4, 5, 6, 7 (omissible)	options for 2 dimensional plot
x-txt = y-txt = z-txt =	(omissible) (omissible) (omissible)	x axis title y axis title z axis title
volume reg vol	(omissible)	The option defines volume for each region for reg mesh. You need volume definitions below this option. Default values are given in input echo in the case of no definition. volume definition. See 5.1.2
iechrl =	72 (default)	Number of maximum column for volume input echo
epsout =	0 (default), 1	If epsout is set to 1, results are plotted into eps files. This eps file is named by replacing the extension into ".eps".
ctmin(i) = ctmax(i) =	(omissible, D=-9999) (omissible, D= 9999)	minimum value for i-th counter maximum value for i-th counter

This tally only scores the energy loss of charged particles. So you cannot get the sum of the energy loss for a specific particle which goes into the tally region by using part = in this tally section. In order to tally the energy loss for each projectile particle going into the tally region, you should define the counter with part = in [counter] section and ctmin, ctmax in this tally section.

6.6 [T - Y i e l d] section

[T - Y i e l d] gives information on produced nuclei informations. Products by neutrons in the energy below `dmax(2)` are not scored, but scored with `e-mode=1`.

Table 6.12: `[t-yield]` parameter (1)

name	value	explanation
`mesh =`	`reg, r-z, xyz`	geometry mesh you need geometry mesh subsection below this option
`special =`	D=0 (omissible)	When `special > 0`, nuclear reactions are repeated more than once in order to increase statistics.
`part =`	`all` (default), particle name	maximum 6 particles in a [t-yield] projectile particle of the reaction
`material =` (next line)	(omissible) `all`, number of materials `2 5 8`	You can specify materials for scoring. all : default (same as no definition) When you set number of materials, define these material numbers in the next line. You can set number of materials by negative. In the case, specified materials are not included for scoring. material numbers
`mother =` (next line)	(omissible) `all`, number of mother nuclei 208Pb Pb	You can specify mother nuclei. all : default (same with no definition) When you set number of mother nuclei, define their mothers in the next line. You can set number of mothers by negative. In this case, specified mothers are not included for scoring. Nucleus if you specify with mass. Without mass, all isotopes of Pb If you want to specify multiple mother groups, use multiple [t-yield] tallies
`nucleus =` (next line)	(omissible) `all`, number of nuclei 208Pb Pb	You can specify output nuclei. all : default (same with no definition) When you set number of nuclei, define their nuclei in the next line. Nucleus If you specify with mass. Without mass, all isotopes of Pb
`unit =`	`1, 2`	1: [1/source] 2: [1/cm^3/source]
`ndata =`	`0`(default), 1	If you set 1, nuclear production cross section data are used for nuclear irradiation in cases of proton induced reactions on α, ^{14}N, ^{16}O targets as shown below.

The following nuclear reactions are included in the available nuclear data for ndata=1

^{4}He$(n,x)^{3}$H	^{14}N$(n,x)^{3}$H	^{14}N$(n,x)^{7}$Be	^{14}N$(n,x)^{11}$Be	^{14}N$(n,x)^{10}$C	^{14}N$(n,x)^{11}$C
^{14}N$(n,x)^{14}$C	^{14}N$(n,x)^{13}$N	^{16}O$(n,x)^{3}$H	^{16}O$(n,x)^{7}$Be	^{16}O$(n,x)^{11}$Be	^{16}O$(n,x)^{10}$C
^{16}O$(n,x)^{11}$C	^{16}O$(n,x)^{14}$C	^{16}O$(n,x)^{15}$C	^{16}O$(n,x)^{13}$N	^{16}O$(n,x)^{16}$N	^{16}O$(n,x)^{14}$O
^{16}O$(n,x)^{15}$O	^{4}He$(p,x)^{3}$H	^{14}N$(p,x)^{7}$Be	^{14}N$(p,x)^{11}$Be	^{14}N$(p,x)^{10}$C	^{14}N$(p,x)^{11}$C
^{14}N$(p,x)^{13}$N	^{14}N$(p,x)^{14}$O	^{16}O$(p,x)^{3}$H	^{16}O$(p,x)^{7}$Be	^{16}O$(p,x)^{11}$Be	^{16}O$(p,x)^{10}$C
^{16}O$(p,x)^{11}$C	^{16}O$(p,x)^{14}$C	^{16}O$(p,x)^{13}$N	^{16}O$(p,x)^{14}$O	^{16}O$(p,x)^{15}$O	

Table 6.13: `[t-yield]` parameter(2)

name	value	explanation

`axis =`	`reg, x, y, z, r,`
	`xy, yz, xz, rz`
	`mass`
	
	`charge`
	`chart`
	
	`dchain`
		x axis for output
		2 dimension
		Mass distribution. If the case nucleus is specified, isotope distribution.
		Charge distribution. Nucleus can not be specified.
		Nucleus chart (x:N, y:Z)。 Nucleus can not be specified
		for dchain-sp output. All isotopes are output only mesh = reg
		
		For this tally, only one `axis` parameter is defined in a `[t-yield] section.`

`file =`	`file name`	Define a file name to output.
resfile =	(omissible, `D=file`)	Define a file name of the past tally in the restart calculation. Even if several `axis` parameters were defined, you should specify only one `resfile`.
`output =`	(omissible) `product` (default) `cutoff`	change the timing of the score. Nuclei produced by nuclear reaction are tallied. Nuclei stopped by energy cutoff are tallied. If nuclei are not transported, this is the same as `product`
`info =`	`0, 1`	With stable nuclei and magic number for chart.
`factor =`	(omissible, D=1.0)	normalization factor
`title =`	(omissible)	title
`angel =`	(omissible)	angel parameters
`2d-type =`	`1, 2, 3, 4, 5, 6, 7` (omissible)	options for 2 dimensional plot

`x-txt =`
`y-txt =`
`z-txt =`
	(omissible)
	(omissible)
	(omissible)
		x axis title
		y axis title
		z axis title

If you specify `output=cutoff`, the parameters of `part`, `mother` are neglected.

Table 6.14: `[t-yield]` parameter (3)

name	value	explanation
`gshow =`	`0` (default), `1, 2, 3, 4`	When mesh=xyz, axis=xy,yz,xz, region border (1), material name (2), region name (3), and LAT number(4) are plotted by the option.
`rshow =`	`0` (default), `1, 2, 3`	When mesh=reg, axis=xy,yz,xz, region border (1), material name (2), and region name (3) are plotted by the option. You need xyz mesh section below this option.

`resol =`
 	
`width =`
	`1` (default)
	
	`0.5` (default)
		The option multiplies region line resolution
		by `resol` times with gshow or rshow option.
		The option defines the line thickness
		for gshow or rshow option.

`volume`
 	
`reg vol`
	(omissible)	The option defines volume for each region
		for reg mesh. You need volume definitions
		below this option.
		Default values are given in input
		echo in the case of no definition.
		volume definition. See 5.1.2

`iechrl =`	`72` (default)	Number of maximum column for volume input echo
`volmat =`	(omissible, D=9)	The option corrects a volume value for each mesh when material is defined by xyz mesh. (0 means no correction) Value of volmat means the number of scans for one side of xyz mesh
`epsout =`	`0` (default), `1`	If epsout is set to 1, results are plotted into eps files. This eps file is named by replacing the extension into ".eps".
`ctmin(i) =` `ctmax(i) =`	(omissible, D=-9999) (omissible, D= 9999)	minimum value for i-th counter maximum value for i-th counter
`trcl =`	(omissible)	coordinate transformation number or definition for r-z or xyz mesh
`gslat =`	1(default), 0	1: show lattice boundary in gshow, 0: no

6.7 [T - P r o d u c t] section

[T - P r o d u c t] tallies particles and nuclei produced by nuclear reaction, decay, and fission, and also tallies source particles. The differences from [t-yield] are that [t-product] does not include the contribution from elastic collisions and you can get the energy distribution and time distribution of produced particles and nuclei. This tally is not available for low energy neutron, photon, and electron. For `e-mode=1`, however, particles and nuclei produced by reactions due to neutron with the library can be obtained.

Table 6.15: `[t-product]` parameter (1)

name	value	explanation
`mesh =`	`reg, r-z, xyz`	geometry mesh you need geometry mesh subsection below this option
`part =`	`all` (default), particle name	maximum 6 particles in a [t-product]
`material =`	(omissible) `all`, number of materials	You can specify materials for scoring. all : default (same as no definition) When you set number of materials, define these material numbers in the next line. You can set number of materials by negative. In the case, specified materials are not included for scoring.
(next line)	`2 5 8`	material numbers
`mother =`	(omissible) `all`, number of mother nuclei	You can specify mother nuclei. all : default (same with no definition) When you set number of mother nuclei, define their mothers in the next line. You can set number of mothers by negative. In this case, specified mothers are not included for scoring.
(next line)	208Pb Pb	Nucleus if you specify with mass. Without mass, all isotopes of Pb If you want to specify multiple mother groups, use multiple [t-product] tallies.
`e-type =`	`1, 2, 3, 4, 5`	energy mesh You need energy mesh subsection below this option.
`t-type =`	`1, 2, 3, 4, 5` (omissible)	time mesh You need time mesh subsection below this option.
`a-type =`	`1, 2, -1, -2`	angle mesh (1, 2 :cos, -1, -2 :degree) You need angle mesh subsection below this option.

Table 6.16: `[t-product]` parameter (2)

name	value	explanation
unit =	1, 2, 3, 4, 5, 6	1: [1/source]
		2: [1/cm^3/source]
		3: [1/MeV/source]
		4: [1/cm^3/MeV/source]
		5: [1/Lethargy/source]
		6: [1/cm^3/Lethargy/source]
	11, 12, 13, 14, 15, 16	11:[1/nsec/source]
		12:[1/cm^3/nsec/source]
		13:[1/MeV/nsec/source]
		14:[1/cm^3/MeV/nsec/source]
		15:[1/Lethargy/nsec/source]
		16:[1/cm^3/Lethargy/nsec/source]
	21, 22, 23, 24, 25, 26	21: [1/SR/source]
		22: [1/cm^3/SR/source]
		23: [1/MeV/SR/source]
		24: [1/cm^3/MeV/SR/source]
		25: [1/Lethargy/SR/source]
		26: [1/cm^3/Lethargy/SR/source]
	31, 32, 33, 34, 35, 36	31:[1/nsec/SR/source]
		32:[1/cm^3/nsec/SR/source]
		33:[1/MeV/nsec/SR/source]
		34:[1/cm^3/MeV/nsec/SR/source]
		35:[1/Lethargy/nsec/SR/source]
		36:[1/cm^3/Lethargy/nsec/SR/source]

“Lethargy ” in `unit = 5, 6, 15, 16, 25, 26, 35 or 36` is a natural logarithmic unit of energy, and defined by $\ln(E_{\mathrm{ref}}/E)$ using a reference energy E_{ref} and a particle’s energy E. If you set these units, you can obtain results per Lethargy, which are calculated by Lethargy widths, $\ln(E_{\mathrm{high}}/E_{\mathrm{low}})$, at each energy bins given in the energy mesh subsection. Here, E_{high} and E_{low} are maximum and minimum values of the energy bins, respectively.

Table 6.17: [t-product] parameter (3)

name	value	explanation
axis =	eng, reg, x, y, z, r, xy, yz, xz, rz t	x axis value of output data 2 dimensional time axis
file =	file name	Define file names as same number of axis
resfile =	(omissible, D=file)	Define a file name of the past tally in the restart calculation. Even if several axis parameters were defined, you should specify only one resfile.
output =	source nuclear (default) nonela elastic decay fission	source particle particles from nuclear reaction including elastic particles from nonelastic collision particles from elastic collision particles from decay particles from fission
factor =	(omissible, D=1.0)	normalization factor
title =	(omissible)	title
angel =	(omissible)	angel parameters
2d-type =	1, 2, 3, 4, 5, 6, 7 (omissible)	options for 2 dimensional plot
x-txt = y-txt = z-txt =	(omissible) (omissible) (omissible)	x axis title y axis title z axis title
gshow =	0 (default), 1, 2, 3, 4	When mesh=xyz, axis=xy,yz,xz, region border (1), material name (2), region name (3), and LAT number(4) are plotted by the option.
rshow =	0 (default), 1, 2, 3	When mesh=reg, axis=xy,yz,xz, region border (1), material name (2), and region name (3) are plotted by the option. You need xyz mesh section below this option.
resol = width =	1 (default) 0.5 (default)	The option multiplies region line resolution by resol times with gshow or rshow option. The option defines the line thickness for gshow or rshow option.
volume reg vol	(omissible)	The option defines volume for each region for reg mesh. You need volume definitions below this option. Default values are given in input echo in the case of no definition. volume definition. See 5.1.2
iechrl =	72 (default)	Number of maximum column for volume input echo

Table 6.18: [t-product] parameter (4)

name	value	explanation
volmat =	(omissible, D=9)	The option corrects a volume value for each mesh when material is defined by xyz mesh. (0 means no correction) Value of volmat means the number of scans for one side of xyz mesh.
epsout =	0 (default), 1	If epsout is set to 1, results are plotted into eps files. This eps file is named by replacing the extension into ".eps".
ctmin(i) =	(omissible, D=-9999)	minimum value for i-th counter
ctmax(i) =	(omissible, D= 9999)	maximum value for i-th counter
trcl =	(omissible)	coordinate transformation number or definition for r-z or xyz mesh
dump =	number of data (omissible)	For mesh=reg, the information is dumped on the file. If dump is negative, data are written by ascii, if positive, by binary.
(next line)	data sequence	define the data sequence.
gslat =	1(default), 0	1: show lattice boundary in gshow, 0: no

In the [t-product] tally, you can use the dump option. If the dump option is set, the meshes of e-type and t-type have only the meaning of the maximum and minimum values. The file in which the dump data are written is the file defined by "file = ". When you use this dump parameter, axis and file are restricted to one axis and one file, and unit is always 1. The normal output of the tally is written on the file which name is "filename" with .cfg. From this file, you can get the information on the total normalization factor. To do so, you had better set one mesh for e-type and t-type.

This [t-product] can tally the source particles. By using this function, you can modify the dump file. You can read a dump file and write the information on a new dump file with some modification by setting the dump parameter and output = source in this tally section, and icntl = 6 in the parameter section.

6.8 [T - D P A] section

[T - D P A] gives DPA (Displacement Per Atom) value. This is the number of displaced atoms per a target atom, and represents the radiation damage in materials irradiated by energetic particles. The result by this tally includes the contribution of Coulomb scattering cross section for the charged particle transportation. DPA by low energy neutron can be also obtained by using libraries. In this case, you must specify the library. If you use `e-mode=1`, you can get the DPA values without the DPA library.

Table 6.19: `[t-dpa]` parameter (1)

name	value	explanation
`mesh =`	`reg, r-z, xyz`	geometry mesh you need geometry mesh subsection below this option
`part =`	`all` (default), particle name	maximum 6 particles in a [t-dpa]
`material =`	(omissible) `all`, number of materials	You can specify materials for scoring. all : default (same as no definition) When you set number of materials, define these material numbers in the next line. You can set number of materials by negative. In the case, specified materials are not included for scoring.
(next line)	`2 5 8`	material numbers
`mother =`	(omissible) `all`, number of mother nuclei	You can specify mother nuclei. all : default (same with no definition) When you set number of mother nuclei, define their mothers in the next line. You can set number of mothers by negative. In this case, specified mothers are not included for scoring.
(next line)	208Pb Pb (mother is not effective in library use)	Nucleus if you specify with mass. Without mass, all isotopes of Pb. If you want to specify multiple mother groups, use multiple [t-dpa] tallies.
`unit =`	`1, 2`	1: [DPA/source*1.e+24] 2: [DPA/source]
`axis =`	`eng, reg, x, y, z, r,` `xy, yz, xz, rz`	x axis value of output data 2 dimensional
`file =`	`file name`	Define file names as same number of axis
resfile =	(omissible, `D=file`)	Define a file name of the past tally in the restart calculation. Even if several `axis` parameters were defined, you should specify only one `resfile`.

Table 6.20: `[t-dpa]` parameter (2)

name	value	explanation
`output =`	`dpa`	total: total DPA value cutoff1: DPA value when energies of charged particles produced by reactions are below cutoff energy (`emin`) cutoff2: DPA value when energies of charged particles transported in materials are below cutoff energy (`emin`) transpt: DPA value when charged particles are transported library: DPA value from neutron library
	`all`	add d, t, ^{3}He, α, and nucleus contributions as PKA, with "simple"
`factor =`	(omissible, D=1.0)	normalization factor
`title =`	(omissible)	title
`angel =`	(omissible)	angel parameters
`2d-type =`	`1, 2, 3, 4, 5, 6, 7` (omissible)	options for 2 dimensional plot
`x-txt =`	(omissible)	x axis title
`y-txt =`	(omissible)	y axis title
`z-txt =`	(omissible)	z axis title
`gshow =`	`0` (default), `1, 2, 3, 4`	When mesh=xyz, axis=xy,yz,xz, region border (1), material name (2), region name (3), and LAT number(4) are plotted by the option.
`rshow =`	`0` (default), `1, 2, 3`	When mesh=reg, axis=xy,yz,xz, region border (1), material name (2), and region name (3) are plotted by the option. You need xyz mesh section below this option.
`resol =`	`1` (default)	The option multiplies region line resolution by `resol` times with gshow or rshow option.
`width =`	`0.5` (default)	The option defines the line thickness for gshow or rshow option.
`volume`	(omissible)	The option defines volume for each region for reg mesh. You need volume definitions below this option. Default values are given in input echo in the case of no definition.
`reg vol`		volume definition. See 5.1.2
`iechrl =`	`72` (default)	Number of maximum column for volume input echo
`volmat =`	(omissible, D=9)	The option corrects a volume value for each mesh when material is defined by xyz mesh. (0 means no correction) Value of volmat means the number of scans for one side of xyz mesh.
`epsout =`	`0` (default), `1`	If epsout is set to 1, results are plotted into eps files. This eps file is named by replacing the extension into ".eps".
`library =`	number of materials (omissible)	Define DPA library for each material. Format is shown later.
`ctmin(i) =`	(omissible, D=-9999)	minimum value for i-th counter
`ctmax(i) =`	(omissible, D= 9999)	maximum value for i-th counter
`trcl =`	(omissible)	coordinate transformation number or definition for r-z or xyz mesh
`gslat =`	1(default), 0	1: show lattice boundary in gshow, 0: no

Format of library specification

```
library =  number of material
   part = proton
   emax = 3000
 mat    fac      lib    mt
   1    1.0      41     445
   2    1.0      42     445
   3    1.0      43     445
 ....   ....     ....   ....
 ....   ....     ....   ....
library =  number of material
   part = neutron
   emax = 3000
 mat    fac      lib    mt
   1    1.0      41     444
   2    1.0      42     444
   3    1.0      43     444
 ....   ....     ....   ....
 ....   ....     ....   ....
```

Define particles for library use by `part =`. Neutron and proton are available in this version. The `emax` defines the maximum energy of data from libraries. If you skip `part` definition, neutron is set. If you skip maximum energy set, the maximum energy of the library is defined. The `mat` is the material number for library use, the `fac` is a normalization factor, the `lib` is material number which contains the library, and the `mt` is DPA record number in the library. In the library made by Dr. Harada, `mt=445` for proton, `mt=444` for neutron. The `fac` and `mt` are omissible. If you skip the `mt` definition, 444 is used. Material number defined by the `lib` should be defined in the `[material]` section. The `y-type` data are assumed for library data reading as shown below.

```
m41     4009.12y 1
m42    13027.12y 1
m43    26054.12y 3.3066d-04  26056.12y 5.2290d-02
       26057.12y 1.2542d-03  26058.12y 1.5963d-04
```

You can change the order "`mat fac lib mt` " like "`mat lib mt fac`". You can use the skip operation `non`.

6.9 [T - L E T] section

By the LET tally, you can get the information on track length and dose as a function of LET(dE/dx) of a certain material. This tally counts an energy loss of charged particles and nuclei, and thus, you must use the Event Generator mode (`e-mode = 1`) if you would like to transport low-energy neutrons.

Table 6.21: `[t-let]` parameters(1)

name	value	explanation
`mesh =`	`reg, r-z, xyz`	geometry mesh you need geometry mesh subsection below this option
`part =`	`all` (default), particle name	maximum 6 particles in a [t-let]
`material =`	(omissible) `all`, number of materials	You can specify materials for scoring. all : default (same as no definition) When you set number of materials, define these material numbers in the next line. You can set number of materials by negative. In the case, specified materials are not included for scoring.
(next line)	`2 5 8`	material numbers
`letmat =`	(omissible)	material id for LET(dE/dx). if omitted, real material is assumed. If you select the material that is not used in your geometry, you have to define its material density in [material] section.
`l-type =`	`1, 2, 3, 4, 5`	LET mesh You need LET mesh subsection below this option It is noted that the LET spectrum may have unnatural peaks when you set a very fine mesh, e.g., 20 meshes per one order of magnitude.
unit =	`1, 2, 3, 4, 5, 6`	1: Track [cm/(keV/μm)/source] 2: Dose [MeV/(keV/μm)/source] 3: Track [cm/ln(keV/μm)/source] 4: Dose [MeV/ln(keV/μm)/source] 5: Track [cm/source] 6: Dose [MeV/source]
	`7, 8, 9, 10, 11, 12`	7: Track [1/cm^2/(keV/μm)/source] 8: Dose [MeV/cm^3/(keV/μm)/source] 9: Track [1/cm^2/ln(keV/μm)/source] 10:Dose [MeV/cm^3/ln(keV/μm)/source] 11:Track [1/cm^2/source] 12:Dose [MeV/cm^3/source]

Table 6.22: [t-let] parameter (2)

name	value	explanation
axis =	let, reg, x, y, z, r, xy, yz, xz, rz	x axis value of output data 2 dimensional
file =	file name	Define file names as same number of axis
resfile =	(omissible, D=file)	Define a file name of the past tally in the restart calculation. Even if several axis parameters were defined, you should specify only one resfile.
factor =	(omissible, D=1.0)	normalization factor
title =	(omissible)	title
angel =	(omissible)	angel parameters
2d-type =	1, 2, 3, 4, 5, 6, 7 (omissible)	options for 2 dimensional plot
x-txt = y-txt = z-txt =	(omissible) (omissible) (omissible)	x axis title y axis title z axis title
gshow =	0 (default), 1, 2, 3, 4	When mesh=xyz, axis=xy,yz,xz, region border (1), material name (2), region name (3), and LAT number(4) are plotted by the option.
rshow =	0 (default), 1, 2, 3	When mesh=reg, axis=xy,yz,xz, region border (1), material name (2), and region name (3) are plotted by the option. You need xyz mesh section below this option.
resol = width =	1 (default) 0.5 (default)	The option multiplies region line resolution by resol times with gshow or rshow option. The option defines the line thickness for gshow or rshow option.
volume reg vol	(omissible)	The option defines volume for each region for reg mesh. You need volume definitions below this option. Default values are given in input echo in the case of no definition. volume definition. See 5.1.2
iechrl =	72 (default)	Number of maximum column for volume input echo
volmat =	(omissible, D=9)	The option corrects a volume value for each mesh when material is defined by xyz mesh. (0 means no correction) Value of volmat means the number of scans for one side of xyz mesh.
epsout =	0 (default), 1	If epsout is set to 1, results are plotted into eps files. This eps file is named by replacing the extension into ".eps".
ctmin(i) = ctmax(i) =	(omissible, D=-9999) (omissible, D= 9999)	minimum value for i-th counter maximum value for i-th counter
trcl =	(omissible)	coordinate transformation number or definition for r-z or xyz mesh
gslat =	1(default), 0	1: show lattice boundary in gshow, 0: no

6.10 [T - S E D] section

Calculation of the probability density of deposition energies in microscopic sites, called as lineal energy *y* or specific energy *z*, is of great importance in estimation of relative biological effectiveness (RBE) of charged particles. However, such microscopic probability densities cannot be directly calculated by *PHITS* simulation using [t-deposit] or [t-heat] tallies, since *PHITS* is designed to simulate particle motions in macroscopic scale, and employs a continuous-slowing-down approximation (CSDA) for calculating the energy loss of charged particles. We therefore introduced a special tally named [t-sed] for calculating the microscopic probability densities using a mathematical function that can instantaneously calculate quantities around trajectories of charged particles. The function was developed on the basis of track structure simulation, considering productions of δ-rays and Auger electrons. Note that the name of " SED " derives from "Specific Energy Distribution". Details of the calculation procedure are given elsewhere.[8, 9)]

Using this tally, we can get information on probability densities of *y* and *z* in water. We can also calculate the probability densities in different materials, although the accuracy has not been checked yet. Similar to [t-let], the dose is only counted in an energy loss of charged particles and nuclei, and thus, we must use the event generator mode (`e-mode = 1`) if we would like to transport low-energy neutrons. The deposition energy in microscopic sites can be expressed by deposit energy ε in MeV, lineal energy y in keV/μm or specific energy *z* in Gy. The definitions of these quantities are given in ICRU Report 36.[10)] Usage of [t-sed] is similar to that of [t-let].

Table 6.23: `[t-sed]` parameters(1)

name	value	explanation
`mesh =`	`reg, r-z, xyz`	geometry mesh you need geometry mesh subsection below this option
`part =`	`all` (default), particle name	maximum 6 particles in a [t-sed]
`material =`	(omissible) `all`, number of materials	You can specify materials for scoring. all : default (same as no definition) When you set number of materials, define these material numbers in the next line. You can set number of materials by negative. In the case, specified materials are not included for scoring.
(next line)	`2 5 8`	material numbers
`letmat =`	(omissible)	material id for LET(dE/dx). if omitted, real material is assumed. If you select the material that is not used in your geometry, you have to define its material density in [material] section.
`se-unit =`	`1, 2, 3`	Unit of deposition energy in microscopic site 1: deposit energy ε in MeV 2: lineal energy *y* in keV/μm 3: specific energy *z* in Gy
`cdiam =`	(omissible, D=1.0)	Diameter of the microscopic site in μm. You can select the value from 0.001 to 2.0.

Table 6.24: [t-sed] parameters(2)

name	value	explanation
se-type =	1, 2, 3, 4, 5	ε, y or z mesh (unit is defined by se-unit). You need energy mesh subsection below this option (specified in ne, emin, emax etc.). If you encounter 'Warning: Z bin is not enough!', then you have to change emin, emax and ne parameters. This warning indicates that the microdosimetric function cannot calculate the y (or z) distribution because of too small range or too poor resolution of its mesh. For example, you have to set se-type =3, emin = 0.01, emax = 10000.0, ne = 60 or more for calculating y distribution for cite diameter = 1 μm. (cdiam = 1.0, se-unit = 2)
unit =	1, 2, 3, 4, 5, 6	1: Track [cm/(keV/μm)/source] 2: Dose [MeV/(keV/μm)/source], corresponding to $y * f(y)$ 3: Track [cm/ln(keV/μm)/source] 4: Dose [MeV/ln(keV/μm)/source], corresponding to $y * d(y)$ 5: Track [cm/source] 6: Dose [MeV/source] The units are for the case of se-unit=2. For se-unit=1 and 3, (keV/μm) is replaced by (MeV) and (Gy), respectively.
axis =	sed, reg, x, y, z, r, xy, yz, xz, rz	x axis value of output data 2 dimensional
file =	file name	Define file names as same number of axis
resfile =	(omissible, D=file)	Define a file name of the past tally in the restart calculation. Even if several axis parameters were defined, you should specify only one resfile.
factor =	(omissible, D=1.0)	normalization factor
title =	(omissible)	title
angel =	(omissible)	angel parameters
2d-type =	1, 2, 3, 4, 5, 6, 7 (omissible)	options for 2 dimensional plot
x-txt = y-txt = z-txt =	(omissible) (omissible) (omissible)	x axis title y axis title z axis title

Table 6.25: `[t-sed]` parameters(3)

name	value	explanation
`gshow =`	`0` (default), `1, 2, 3, 4`	When mesh=xyz, axis=xy,yz,xz, region border (1), material name (2), region name (3), and LAT number(4) are plotted by the option.
`rshow =`	`0` (default), `1, 2, 3`	When mesh=reg, axis=xy,yz,xz, region border (1), material name (2), and region name (3) are plotted by the option. You need xyz mesh section below this option.

`resol =`
 	
`width =`
	`1` (default)
	
	`0.5` (default)
		The option multiplies region line resolution
		by `resol` times with gshow or rshow option.
		The option defines the line thickness
		for gshow or rshow option.

`volume`
 	
`reg vol`
	(omissible)	The option defines volume for each region
		for reg mesh. You need volume definitions
		below this option.
		Default values are given in input
		echo in the case of no definition.
		volume definition. See 5.1.2

`iechrl =`	`72` (default)	Number of maximum column for volume input echo
`volmat =`	(omissible, D=9)	The option corrects a volume value for each mesh when material is defined by xyz mesh. (0 means no correction) Value of volmat means the number of scans for one side of xyz mesh.
`epsout =`	`0` (default), `1`	If epsout is set to 1, results are plotted into eps files. This eps file is named by replacing the extension into ".eps".
`ctmin(i) =` `ctmax(i) =`	(omissible, D=-9999) (omissible, D= 9999)	minimum value for i-th counter maximum value for i-th counter
`trcl =`	(omissible)	coordinate transformation number or definition for r-z or xyz mesh
`gslat =`	1(default), 0	1: show lattice boundary in gshow, 0: no

6.11 [T - T i m e] section

[T - T i m e] gives number of energy cut off and escape particles by the time mesh (nsec).

Table 6.26: `[t-time]` parameter (1)

name	value	explanation
`mesh =`	`reg, r-z, xyz`	geometry mesh you need geometry mesh subsection below this option
`part =`	`all` (default), particle name	maximum 6 particles in a [t-time]
`material =`	(omissible) `all`, number of materials	You can specify materials for scoring. all : default (same as no definition) When you set number of materials, define these material numbers in the next line. You can set number of materials by negative. In the case, specified materials are not included for scoring.
(next line)	`2 5 8`	material numbers
`t-type =`	`1, 2, 3, 4, 5`	time mesh You need time mesh subsection below this option
`e-type =`	`1, 2, 3, 4, 5`	energy mesh You need energy mesh subsection below this option
`unit =`	`1, 2, 3, 4`	1: [1/source] 2: [1/nsec/source] 3: [1/nsec/cm^3/source] 4: [1/nsec/cm^3/MeV/source]
`axis =`	`eng, reg, x, y, z, r,` `xy, yz, xz, rz`	x axis value of output data 2 dimensional
`file =`	`file name`	Define file names as same number of axis
resfile =	(omissible, `D=file`)	Define a file name of the past tally in the restart calculation. Even if several `axis` parameters were defined, you should specify only one `resfile`.

`output =`	`all`
	`cutoff`
	`escape`
		energy cut off and escape particles
		energy cut off particles
		escape particles

`factor =`	(omissible, D=1.0)	normalization factor
`title =`	(omissible)	title
`angel =`	(omissible)	angel parameters
`2d-type =`	`1, 2, 3, 4, 5, 6, 7` (omissible)	options for 2 dimensional plot

Table 6.27: [t-time] parameter (2)

name	value	explanation
x-txt = y-txt = z-txt =	(omissible) (omissible) (omissible)	x axis title y axis title z axis title
gshow =	0 (default), 1, 2, 3, 4	When mesh=xyz, axis=xy,yz,xz, region border (1), material name (2), region name (3), and LAT number(4) are plotted by the option.
rshow =	0 (default), 1, 2, 3	When mesh=reg, axis=xy,yz,xz, region border (1), material name (2), and region name (3) are plotted by the option. You need xyz mesh section below this option.
resol = width =	1 (default) 0.5 (default)	The option multiplies region line resolution by resol times with gshow or rshow option. The option defines the line thickness for gshow or rshow option.
volume reg vol	(omissible)	The option defines volume for each region for reg mesh. You need volume definitions below this option. Default values are given in input echo in the case of no definition. volume definition. See 5.1.2
iechrl =	72 (default)	Number of maximum column for volume input echo
volmat =	(omissible, D=9)	The option corrects a volume value for each mesh when material is defined by xyz mesh. (0 means no correction) Value of volmat means the number of scans for one side of xyz mesh
epsout =	0 (default), 1	If epsout is set to 1, results are plotted into eps files. This eps file is named by replacing the extension into ".eps".
ctmin(i) = ctmax(i) =	(omissible, D=-9999) (omissible, D= 9999)	minimum value for i-th counter maximum value for i-th counter
trcl =	(omissible)	coordinate transformation number or definition for r-z or xyz mesh
dump = (next line)	number of data (omissible) data sequence	For mesh=reg, the information is dumped on the file. If dump is negative, data is written by ascii, if positive, by binary. define the data sequence.
gslat =	1(default), 0	1: show lattice boundary in gshow, 0: no

You can obtain energy spectra of the energy cut off and escape particles by the [t-time] tally. Especially, [t-time] is the only tally to give energy spectra of the particles which can not be transported in $PHITS$, since they are assumed as cut off particles in the code.

In [t-time] tally, you can use the dump option only with output = cutoff. If the dump option is set, the meshes of e-type and t-type have only the meaning of the maximum and minimum values, and unit is set to be 1. The file in which the dump data are written is the file defined by "file = ". When you use this dump parameter, axis and file are restricted to one axis and one file. The normal output of the tally is written on the file which name is "filename" with .cfg. From this file, you can get the information on the total normalization factor. To do so, you had better set one mesh for e-type and t-type.

By this dump option, you can create similar files to ncut, gcut and pcut files for the sequential calculations of the other transport code.

6.12 [T - S t a r] section

[T - S t a r] gives star density which is the distribution of the nuclear reactions. Reactions for electron by libraries are not included.

Table 6.28: [t-star] parameter (1)

name	value	explanation
mesh =	reg, r-z, xyz	geometry mesh you need geometry mesh subsection below this option
part =	all (default), particle name	maximum 6 particles in a [t-star] projectile particle of the reaction
material =	(omissible) all, number of materials	You can specify materials for scoring. all : default (same as no definition) When you set number of materials, define these material numbers in the next line. You can set number of materials by negative. In the case, specified materials are not included for scoring.
(next line)	2 5 8	material numbers
mother =	(omissible) all, number of mother nuclei	You can specify mother nuclei. all : default (same with no definition) When you set number of mother nuclei, define their mothers in the next line. You can set number of mothers by negative. In this case, specified mothers are not included for scoring.
(next line)	208Pb Pb	Nucleus if you specify with mass. Without mass, all isotopes of Pb. If you want to specify multiple mother groups, use multiple [t-star] tallies
nucleus =	(omissible) all, number of nuclei	You can specify output nuclei. all : default (same with no definition) When you set number of nuclei, define their nuclei in the next line.
(next line)	208Pb Pb	Nucleus if you specify with mass. Without mass, all isotopes of Pb.
e-type =	1, 2, 3, 4, 5	energy mesh You need energy mesh subsection below this option.
t-type =	1, 2, 3, 4, 5 (omissible)	time mesh You need time mesh subsection below this option
unit =	1, 2	1: [1/cm^3/source] 2: [1/cm^3/MeV/source]
axis =	eng, reg, x, y, z, r, xy, yz, xz, rz	x axis value of output data 2 dimensional
file =	file name	Define file names as same number of axis.

Table 6.29: `[t-star]` parameter (2)

name	value	explanation
resfile =	(omissible, `D=file`)	Define a file name of the past tally in the restart calculation. Even if several `axis` parameters were defined, you should specify only one `resfile`.

`output =`	`all`
	`decay`
	`elastic`
	`nuclear`
	`fission`
	`absorption`
	`heavyion`
		star density for all reactions
		star density for decay reaction
		star density for elastic reaction
		star density for non-elastic + Hydrogen + HI
		star density for fission
		star density for absorption
		star density for Heavy Ion reaction

`factor =`	(omissible, D=1.0)	normalization factor
`title =`	(omissible)	title
`angel =`	(omissible)	angel parameters
`2d-type =`	`1, 2, 3, 4, 5, 6, 7` (omissible)	options for 2 dimensional plot

`x-txt =`
`y-txt =`
`z-txt =`
	(omissible)
	(omissible)
	(omissible)
		x axis title
		y axis title
		z axis title

`gshow =`	`0` (default), `1, 2, 3, 4`	When mesh=xyz, axis=xy,yz,xz, region border (1), material name (2), region name (3), and LAT number(4) are plotted by the option.
`rshow =`	`0` (default), `1, 2, 3`	When mesh=reg, axis=xy,yz,xz, region border (1), material name (2), and region name (3) are plotted by the option. You need xyz mesh section below this option.

`resol =`
 	
`width =`
	`1` (default)
	
	`0.5` (default)
		The option multiplies region line resolution
		by `resol` times with gshow or rshow option.
		The option defines the line thickness
		for gshow or rshow option.

`volume`
 	
`reg vol`
	(omissible)	The option defines volume for each region for reg mesh.
		You need volume definitions below this option.
		Default values are given in input echo
		in the case of no definition.
		volume definition. See 5.1.2

`iechrl =`	`72` (default)	Number of maximum column for volume input echo
`volmat =`	(omissible, D=9)	The option corrects a volume value for each mesh when material is defined by xyz mesh. (0 means no correction) Value of volmat means the number of scans for one side of xyz mesh
`epsout =`	`0` (default), `1`	If epsout is set to 1, results are plotted into eps files. This eps file is named by replacing the extension into ".eps".
`ctmin(i) =` `ctmax(i) =`	(omissible, D=-9999) (omissible, D= 9999)	minimum value for i-th counter maximum value for i-th counter
`trcl =`	(omissible)	coordinate transformation number or definition for r-z or xyz mesh
`gslat =`	1(default), 0	1: show lattice boundary in gshow, 0: no

6.13 [T - D c h a i n] section

This tally is used for generating input files for DCHAIN-SP. Figure 6.4 illustrates the flowchart of the connection calculation between *PHITS* and DCHAIN-SP.

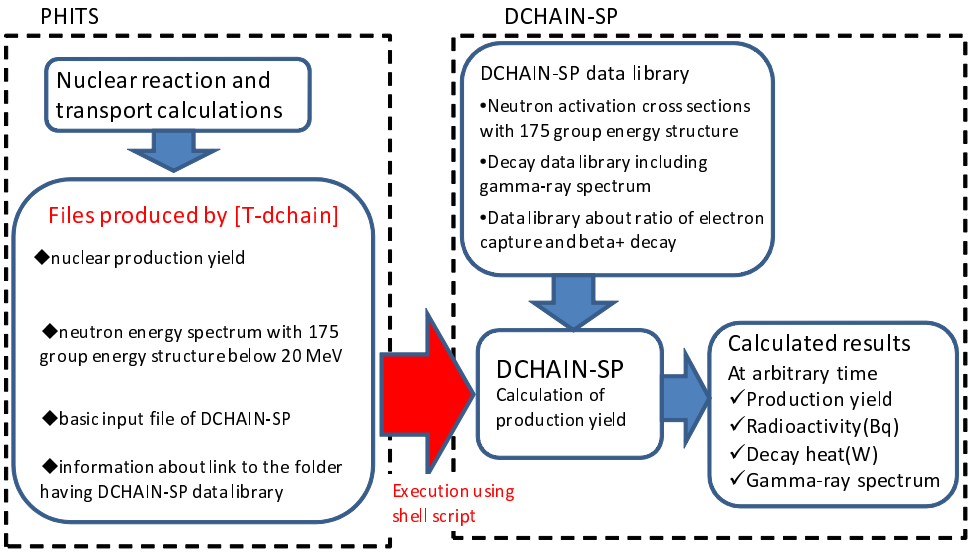


Figure 6.4: Concept of the connection calculation between *PHITS* and DCHAIN-SP.

In the *PHITS* calculation, `[t-dchain]` automatically creates `[t-track]` and `[t-yield]` as well as the input file of DCHAIN-SP. The `[t-track]` tally calculates the neutron energy spectra below 20 MeV with 175-energy-group structure. The `[t-yield]` tally calculates the nuclear production yields by protons, heavy-ions, mesons and neutrons with energies above 20 MeV.

In the DCHAIN-SP calculation, the neutron energy spectra are multiplied with the activation cross section contained in the DCHAIN-SP data library. Then, the total activations are estimated by adding these results and those directly calculated by *PHITS* using the `[t-yield]` tally. After that, DCHAIN-SP evaluates radioactivity, nuclide, decay heat and gamma energy spectrum at irradiation and cooling time. Please see the "\phits\recommendation\dchain" folder in more detail. You have to refer the document[4] below when you use the results obtained by this tally in your publications.

[4] Tetsuya Kai, et al., "DCHAIN-SP 2001: High Energy Particle Induced Radioactivity Calculation Code" , JAEA-Data/Code-2001-016 (2001) in Japanese

Table 6.30: `[t-dchain]` parameter

name	value	explanation
mesh =	reg	Geometry mesh (Currently ONLY region mesh `reg` can be specified). You need to write the geometry mesh subsection below this option. (`reg` = cell number)
file =	file name	Input file name of DCHAIN-SP. You can use any extention except for `.dtrk, .dyld .dout`.
title =	(omissible)	Title.
timeevo =	number	Number of irradiation and cooling steps in DCHAIN-SP calculation.
(next line)	time factor	Time step of irradiation and/or cooling. Normalized factor for beam intensity. Time should be calculated from the end of last step, not from the start of the first irradiation. The allowable units are seconds, `s`, minutes, `m`, hours, `h`, days, `d` and years `y`. One (or more) blank character must be placed between the number indicating the time and the units. *See the example of input for `[T-Dchain]` tally in List 6.1.
outtime =	number	Number of output timing in the DCHAIN-SP calculation.
(next line)	time	Output timing. If a positive value is given as output timing, it is calculated from the start of the first irradiation step. If a negative value is given as output timing, it is calculated from the end of last irradiation step. The format for specifying the timing is the same as that for `timeevo`, except for this positive and negative rule. You cannot specify timing after all steps defined in `timeevo` is finished.
amp =	(omissible,D=1.0)	Power of source (source/second)
target =	0, 1 (omissible,D=0)	OFF:0 need not to write information of `target`. Information of `target` is automatically determined from `[Material]`, `[Cell]`, and `[Volume]` sections. ON:1 need to write information of `target`. *If you want to add nuclides that are not defined in `[Material]` section in the DCHAIN-SP calculation, and/or you do not set the volume in `[Volume]` section, you should write `target=1` and give the related information in List 6.2. *See the example of target subsection for `target=1` in List 6.2.

List 6.1 ● Example of input for [T-Dchain] tally.

```
 1:        mesh = reg                    <-region mesh
 2:         reg = 100                    <-cell number
 3:        file = testDC.spd             <-file name of DCHAIN-SP input
 4:       title = [t-dchain] test calc.
 5:         amp = 1.0E12                 <-source power (source/sec)
 6:
 7:     timeevo = 4                      <-number of irradiation and cooling steps
 8:              3.0 h  1.0              <-irradiation for 3 hours
 9:              2.0 h  0.0              <-cooling for 2 hours
10:              3.5 h  1.0              <-irradiation for 3.5 hours
11:             15.5 h  0.0              <-cooling for 15.5 hours
12:
13:     outtime = 3                      <-number of output timing
14:              3.0 h                   <-3 hours later from the 1st irradiation start time
15:             -1.0 h                   <-1 hour later from the end of the last irradiation step
16:             -3.0 h                   <-3 hour later from the end of the last irradiation step
```

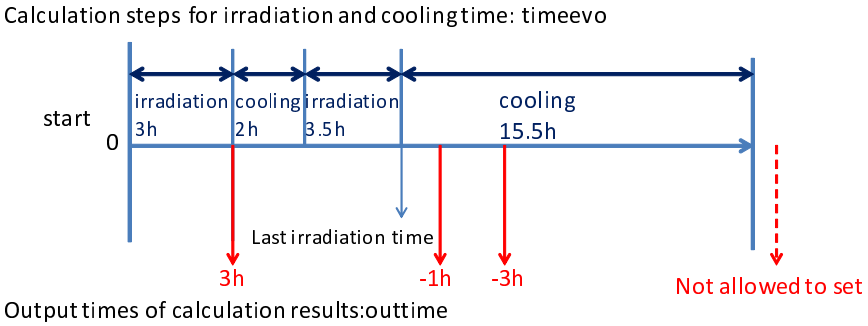


Figure 6.5: Relation between steps for irradiation and cooling time and output time.

List 6.2 ● Example for the setting of target material compositions and volumes for target=1.

```
          .......
:      target = 1                      <-target material composition ON
:      non    reg    vol               <-omissible
:        1     1   8000.0              <-serial number, cell number, volume
:     tg-list = 2                      <-number of the nuclides
:             H-1   6.689E-02          <-Element ID, Atomic Number,
:             O-16  3.345E-02            and Density of the atom (10^24/cm^3)
:       2     2   2000.0              <-serial number, cell number, volume
:     tg-list = 1                      <-Number of the nuclides
:            Fe-56  8.385E-02
```

The symbol of the chemical element has to be connected with atomic number by the character '-'for indicating an isotope.

Important notices for using [t-dchain]:

- Only one [T-Dchain] tally per *PHITS* input file is allowed.
- The following parameters must be defined in the [Parameters] section:
  - jmout=1: display the atomic number density of materials.
  - file(21): set the place of the data folder for DCHAIN-SP.
- You have to define the volume of each tally region in the [Volume] section.

Files generated by [t-dchain] are listed below.

- Basic input file of DCHAIN-SP: file name is set in [T-dchain] tally.

- Neutron energy spectrum with 175 group energy structure below 20 MeV calculated by created `[t-track]`: “n.flux” . When more than two regions are set, each file name is n.flux_01, n.flux_02, $\cdots$, f.flux_XX.
- Nuclear production yields calculated by created `[t-yield]`: “nmtc_yield” .
- Information about the link to the folder having DCHAIN-SP data library: “dch_link.dat” .
- Files for restart calculation of *PHITS* : `***.dtrk, ***.dyld, ***_err.dyld, ***.dout`

6.14 [T-Userdefined] section

This tally is used for estimating and outputting the physical quantities that cannot be calculated by the other tallies. In order to use this tally, you have to change "usrtally.f" and re-compile *PHITS* by yourself.

When `[T-Userdefined]` is defined in your input file, *PHITS* calls subroutine "usrtally" at every moment in the *PHITS* simulation, namely at the same timing for calling the dumpall option. A subroutine for output all information is written in the default "usrtally.f" . You can output only your required information by revising the file.

In `[T-Userdefined]`, you can use two parameters, one is `file` for specifying a file name, and the other is udtpara*i* ($i = 0 - 9$) for specifying a numerical value. These parameters can be used in the subroutine "usrtally" without recompiling *PHITS* .

Table 6.31: `[t-userdefined]` parameter

name	value	explanation
file =	file name	Output file names used in `subroutine usrtally`. The maximum number of files that you can use is 50. (Device numbers of these files are from 151 to 200.)
udtpara*i* = ($i = 0 - 9$)	(omissible)	Parameters used in `subroutine usrtally`. You can define from udtpara0 to udtpara9.

List 6.3 ● Example of input for `[T-Userdefined]` tally.

```
1:        file = output1.dat            <-its device number is iudtf(1)=151
2:        file = output2.dat            <-its device number is iudtf(2)=152
3:    udtpara0 = 10.0                   <-udtpara(0)
4:    udtpara3 = 20.0                   <-udtpara(3)
```

You can use the following parameters in `subroutine usrtally`.

(1) NCOL:
This is an intrinsic variable in the program and denotes identification of process.

NCOL
- 1 : start of calculation
- 2 : end of calculation
- 3 : end of a batch
- 4 : source
- 5 : detection of geometry error
- 6 : recovery of geometry error
- 7 : termination by geometry error
- 8 : termination by weight cut-off
- 9 : termination by time cut-off
- 10 : geometry boundary crossing
- 11 : termination by energy cut-off
- 12 : termination by escape or leakage
- 13 : (n,x) reaction
- 14 : (n,n'x) reaction
- 15 : sequential transport only for tally

(2) npe, me:
These are the number of used PEs (Processor Elements) and ID number of each processor, respectively, in the distributed-memory parallel computing.

(3) ipomp, npomp:
These are ID number of each core and the total number of used cores, respectively, in the shared memory parallel computing.

(4) iusrtally:
This is a parameter to control whether `subroutine usrtally` is used or not. If `[t-userdefined]` is defined in an input file, this parameter is set to be 1.

(5) iudtf(50):
These are device numbers of output files defined with `file=`. For example, if there is the earliest file defined in `[t-userdefined]`, its device number is `iudtf(1)=151`.

(6) udtpara(0:9):
These correspond to parameters defined with `udtpara`i ($i = 0 - 9$). For example, udtpara(0) equals to `udtpara`0.

(7) NOCAS, NOBCH, RCASC, RSOUIN:

NOCAS	:	current history number in this batch
NOBCH	:	current batch number
RCASC	:	real number of NOCAS+maxcas*(NOBCH-1)
RSOUIN	:	sum of the weight of source particle

(8) NO, IDMN, ITYP, KTYP, JTYP, MTYP, RTYP, OLDWT:

NO	:	cascade id in this history
IDMN	:	material id
ITYP	:	particle type
KTYP	:	particle kf-code
JTYP	:	charge number of the particle
MTYP	:	baryon number of the particle
RTYP	:	rest mass of the particle (MeV)
OLDWT	:	wight of the particle at (x,y,z)

(9) QS:
This is dE/dx for electrons at (x,y,z).

(10) IBLZ1, IBLZ2, ILEV1, ILEV2:

IBLZ1	:	cell id at (x,y,z)
IBLZ2	:	cell id after crossing
ILEV1	:	level structure id of the cell at (x,y,z)
ILEV2	:	level structure id of the cell after crossing

(a) ILAT1:
This is a variable of level structure of cell.

(b) ILAT2:
This is a variable of level structure of cell.

(11) COSTH, UANG(1), UANG(2), UANG(3), NSURF:

COSTH	:	cosine of an angle of incidence in a surface crossing
UANG(1,2,3)	:	x,y,z component of a normal vector of its surface, respectively
NSURF	:	internal number of the surface (This is different from the surface number defined in the `[surface]` section.)

(12) NAME, NCNT(1), NCNT(2), NCNT(3):

NAME	:	collision number of the particle
NCNT(1,2,3)	:	values of counter 1, 2, and 3

(13) WT, U, V, W:

WT	:	wight of the particle at (xc,yc,zc)
U, V, W	:	unit vector of momentum of the particle

(14) E, T, X, Y, Z:

E	:	energy of the particle at (x,y,z) (MeV)
T	:	time of the particle at (x,y,z) (nsec)
X, Y, Z	:	position coordinate of the preceding event point (cm)

(15) EC, TC, XC, YC, ZC:

EC	:	energy of the particle at (xc,yc,zc) (MeV)
TC	:	time of the particle at (xc,yc,zc) (nsec)
XC, YC, ZC	:	position coordinate of the particle (cm)

(16) SPX, SPY, SPZ:

SPX, SPY, SPZ	:	unit vector of spin direction of the particle

(17) NZST:
This is charge state of the particle.

(18) NCLSTS:
This variable means the number of produced particle and nucleus.

(a) MATHZ, MATHN, JCOLL, KCOLL:

MATHZ	:	Z number of the mother nucleus
MATHN	:	N number of the mother nucleus
JCOLL	:	reaction type id1
KCOLL	:	reaction type id2

JCOLL and KCOLL indicate the following meaning.

JCOLL

0	:	nothing happen
1	:	Hydrogen collisions
2	:	Particle Decays
3	:	Elastic collisions
4	:	High Energy Nuclear collisions
5	:	Heavy Ion reactions
6	:	Neutron reactions by data
7	:	Photon reactions by data
8	:	Electron reactions by data
9	:	Proton reactions by data
10	:	Neutron event mode
11	:	delta ray production

KCOLL

0	:	normal
1	:	high energy fission
2	:	high energy absorption
3	:	low energy n elastic
4	:	low energy n non-elastic
5	:	low energy n fission
6	:	low energy n absorption

(b) ICLUSTS, JCLUSTS, QCLUSTS, JCOUNT:
These variables have a array and denote the information on the produced particle and nucleus.

ICLUSTS kind of particle

0	:	nucleus
1	:	proton
2	:	neutron
3	:	pion
4	:	photon
5	:	kaon
6	:	muon
7	:	others

JCLUSTS(i)

i = 0	:	angular momentum
= 1	:	proton number
= 2	:	neutron number
= 3	:	ityp
= 4	:	status of the particle 0: real, <0 : dead
= 5	:	charge number
= 6	:	baryon number
= 7	:	kf code

QCLUSTS(i)

i = 0	:	impact parameter
= 1	:	px (GeV/c)
= 2	:	py (GeV/c)
= 3	:	pz (GeV/c)
= 4	:	$e_{tot} = \sqrt{p^2 + m^2}$ (GeV)
= 5	:	rest mass (GeV)
= 6	:	excitation energy (MeV)
= 7	:	kinetic energy (MeV)
= 8	:	weight
= 9	:	time (nsec)
= 10	:	x coordinate (cm)
= 11	:	y coordinate (cm)
= 12	:	z coordinate (cm)

6.15 [T - G s h o w] section

[T - G s h o w] gives graphical geometry output for region boundary by xyz mesh. You can obtain these results without transport calculations with `icntl =7` option in the [parameters] section.

Table 6.32: `[t-gshow]` parameter

name	value	explanation
`mesh =`	`xyz`	geometry mesh, only xyz mesh you need geometry mesh subsection below this option
`axis =`	`xy, yz, xz`	2 dimensional
`file =`	`file name`	Define file names as same number of axis

`output =`	`1`
	`2`
	`3`
	`4`
	`5`
	`6`
	`7`
	`8`
		region boundary
		region boundary + material color
		region boundary + material name
		region boundary + material color + material name
		region boundary + region name
		region boundary + material color + region name
		region boundary + LAT number
		region boundary + material color + LAT number

`resol =`	`1` (default)	The resolution in displaying region lines is multiplied by this value.
`width =`	`0.5` (default)	The thickness of displayed region lines.
`title =`	(omissible)	title
`angel =`	(omissible)	angel parameters
`x-txt =`	(omissible)	x axis title
`y-txt =`	(omissible)	y axis title
`epsout =`	`0` (default), `1`	If epsout is set to 1, results are plotted into eps files. This eps file is named by replacing the extension into ".eps".
`trcl =`	(omissible)	coordinate transformation number or definition for r-z or xyz mesh
`gslat =`	1(default), 0	1: show lattice boundary in gshow, 0: no

`output=7,8` can be used only when cells in bottom level are the lattice themselves, and they give lattice number in the format as (4,1,2). For example, Fig. 4.17 in Sec. 4.7.4 is generated by the input shown below.

List 6.4 ● `[t-gshow]` example

```
 1:   [ T - gshow ]
 2:         mesh = xyz
 3:       x-type = 2
 4:           nx = 100
 5:         xmin = -10.
 6:         xmax =  10
 7:       y-type = 1
 8:           ny = 1
 9:                -5.0 5.0
10:       z-type = 2
11:        nz    = 100
12:        zmin  = -10.
13:        zmax  =  10.
14:         axis = xz
15:       output = 8
16:         file = cell-example6.dat
17:       epsout = 1
```

6.16 [T - R s h o w] section

[T - R s h o w] gives graphical geometry output for region boundary with color plot region in proportion to physical quantity of the region. Usually, the results obtained by other *PHITS* calculation using the reg mesh are used as the input data for this value of physical quantity. You must run *PHITS* with `icntl =9` option in the [parameters] section, in order to execute this tally.

You can give color variation by the linear scale or the log scale by the *ANGEL* parameter, zlog or zlin. Default is zlin.

Table 6.33: `[t-rshow]` parameter

name	value	explanation
`mesh =`	`xyz`	geometry mesh, only xyz mesh you need geometry mesh subsection below this option
`axis =`	`xy, yz, xz`	2 dimensional
`file =`	`file name`	Define file names as same number of axis

`output =`	`1`
	`2`
	`3`
	`4`
		region boundary
		region boundary + material name
		region boundary + region name
		region boundary + LAT number

`resol =`	`1` (default)	The resolution in displaying region lines is multiplied by this value.
`width =`	`0.5` (default)	The thickness of displayed region lines.
`title =`	(omissible)	title
`angel =`	(omissible)	angel parameters
`x-txt =`	(omissible)	x axis title
`y-txt =`	(omissible)	y axis title
`z-txt =`	(omissible)	z axis title
`reg =` value reg val		region definition value definition with same format as volume definition see section 5.1.2
`iechrl =`	`72` (default)	Number of maximum column for volume input echo
`epsout =`	`0` (default), `1`	If epsout is set to 1, results are plotted into eps files. This eps file is named by replacing the extension into ".eps".
`trcl =`	(omissible)	coordinate transformation number or definition for r-z or xyz mesh
`gslat =`	1(default), 0	1: show lattice boundary in gshow, 0: no

For example, you can obtain Fig. 6.6 by the `[t-rshow]` tally shown below from the example (6) in Sec. 4.7.4.

List 6.5 ● `[t-rshow]` example

```
 1:   [ T - rshow ]
 2:        mesh = xyz
 3:      x-type = 2
 4:          nx = 100
 5:        xmin = -10.
 6:        xmax =  10.
 7:      y-type = 1
 8:          ny = 1
 9:               -5.0  5.0
10:      z-type = 2
11:          nz = 100
12:        zmin = -10.
13:        zmax =  10.
14:        axis = xz
15:      output = 1
16:        file = cell-example6-rshow.dat
17:      epsout = 1
18:         reg = (201<101[-1 1 0]<1) (201<101[0 1 0]<1) (201<101[1 1 0]<1)
19:               (201<101[-1 0 0]<1) (201<101[0 0 0]<1) (201<101[1 0 0]<1)
20:               (201<101[-1 -1 0]<1) (201<101[0 -1 0]<1) (201<101[1 -1 0]<1)
21:       value
22:      non     reg      val     # reg definition
23:       1  1000001   1.0000E+00 # ( 201 < 101[ -1 1 0 ] < 1 )
24:       2  1000002   2.0000E+00 # ( 201 < 101[ 0 1 0 ] < 1 )
25:       3  1000003   3.0000E+00 # ( 201 < 101[ 1 1 0 ] < 1 )
26:       4  1000004   4.0000E+00 # ( 201 < 101[ -1 0 0 ] < 1 )
27:       5  1000005   5.0000E+00 # ( 201 < 101[ 0 0 0 ] < 1 )
28:       6  1000006   6.0000E+00 # ( 201 < 101[ 1 0 0 ] < 1 )
29:       7  1000007   7.0000E+00 # ( 201 < 101[ -1 -1 0 ] < 1 )
30:       8  1000008   8.0000E+00 # ( 201 < 101[ 0 -1 0 ] < 1 )
31:       9  1000009   9.0000E+00 # ( 201 < 101[ 1 -1 0 ] < 1 )
```

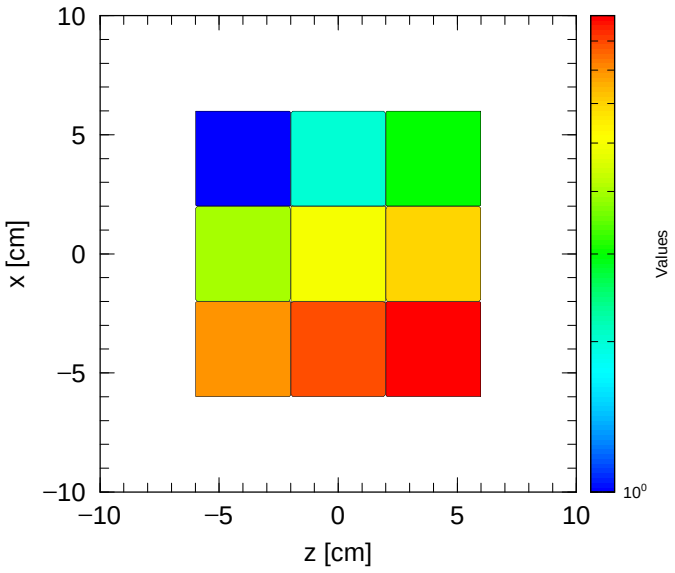


Figure 6.6: Example of `[t-rshow]`.

6.17 [T - 3 D s h o w] section

[T - 3 D s h o w] gives a graphical geometry output by 3 dimensional view. You can execute this tally with `icntl =11` option in the [parameters] section without transport calculations.

Table 6.34: `[t-3dshow]` parameter (1)

name	value	explanation

`output =`	`0`
	`1`
	`2`
	`3` (default)
		draft
		only region boundary
		without region boundary
		region boundary + color

`material =`	(omissible) `all`, number of materials (next line)	You can specify materials for display. all : default (same as no definition) When you set number of materials, define these material numbers in the next line. You can set number of materials by negative. In the case, specified materials are not included for display. `2 5 8`	material numbers

`x0 =`
`y0 =`
`z0 =`
	(D=0.0)
	(D=0.0)
	(D=0.0)
		Coordinates of original point for view point
		and light source. Center of screen is defined
		by this point and view point

`e-the =`
`e-phi =`
`e-dst =`
	(D=80)
	(D=140)
	(D=w-dst*10)
		view point angle θ(degree) with z axis
		azimuthal angle for view point ϕ(degree) with x axis
		distance between view point and the origin (cm)

`l-the =`
`l-phi =`
`l-dst =`
	(D=e-the)
	(D=e-phi)
	(D=e-dst)
		light source angle θ(degree) with z axis
		azimuthal angle for light source ϕ(degree) with x axis
		distance between light source and the origin (cm)

`w-wdt =`
`w-hgt =`
`w-dst =`
 	
`w-mnw =`
`w-mnh =`
`w-ang =`
	(D=100)
	(D=100)
	(D=200)
	
	(D=100)
	(D=100)
	(D=0.0)
		width of screen frame (cm)
		height of screen frame(cm)
		screen frame distance from the origin (cm)
		A straight line drawn between the center of screen frame
		and the origin crosses screen surface vertically,
		and passes through the view point.
		number of mesh for horizontal direction
		number of mesh for vertical direction
		angle of frame (degree)

`heaven =`	(D=y)	topside direction; set x, -x, y, -y, z, or -z
`mirror =`	(D=0)	=-1; mirror transformation in horizontal direction

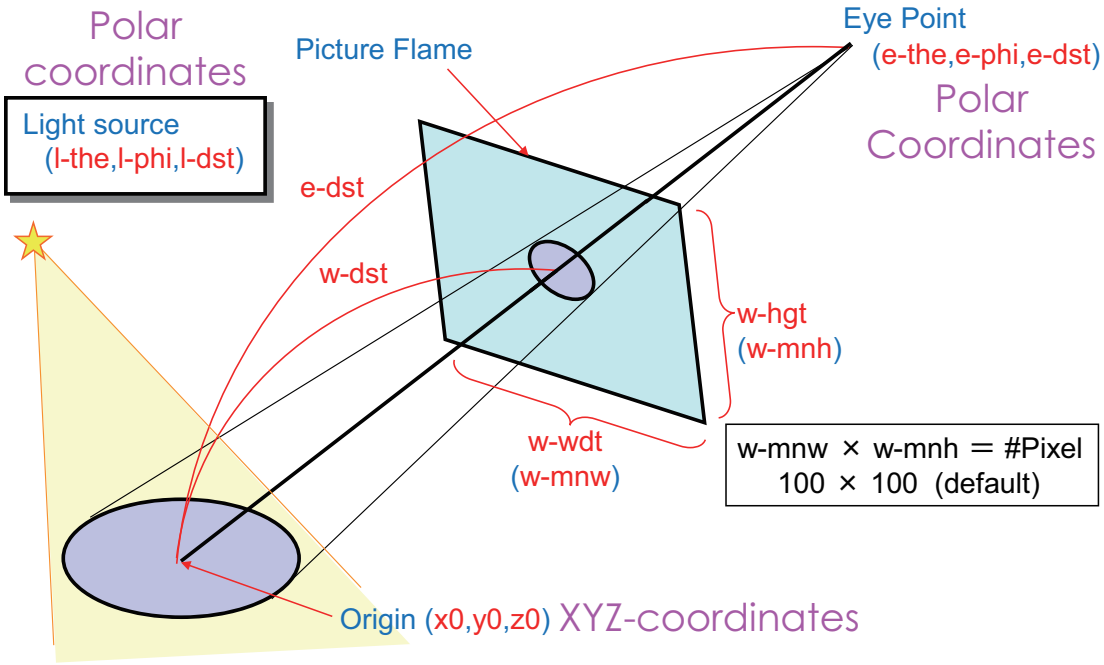


Figure 6.7: 3dshow tally: origin (`x0,y0,z0`), eye point (`e-the,e-phi,e-dst`)、light source (`l-the,l-phi,l-dst`), and picture flame (`w-wdt,w-hgt,w-dst`).

Table 6.35: [t-3dshow] parameter (2)

name	value	explanation
line =	0 (default), 1	When output = 1, 3 0: material boundary + surface boundary 1: material boundary + surface boundary + region boundary
r-out =	(D=50000)	radius of outer void including view point, and light source(cm)
shadow =	(D=0)	shadow level (0:no shadow, 2 is recommended)
bright =	(D=0.8)	brightness limit (1:max, 0:no brightness)
dark =	(D=0.2)	darkness limit (1:no darkness, 0:max)
box =	(D=0)	number of penetration box, maximum 5
box	10 numbers	box definition (see below)
matinbox =	(omissible) all, number of materials	materials in the box for display all : default (same with no definition) When you set number of materials, define these material numbers in the next line. You cannot set number of materials by negative.
(next line)	2 5 8	material numbers
reginbox =	(omissible) all, region numbers	regions in the box for display all : default (same with no definition) If the matinbox is defined for this region, this region is not displayed.
resol =	1 (default)	The resolution in displaying region lines is multiplied by this value.
width =	0.5 (default)	The thickness of displayed region lines.
file =	file name	Define file names as same number of axis
title =	(omissible)	title
angel =	(omissible)	angel parameters
x-txt =	(omissible)	x axis title
y-txt =	(omissible)	y axis title
z-txt =	(omissible)	z axis title
epsout =	0 (default), 1	If epsout is set to 1, results are plotted into eps files. This eps file is named by replacing the extension into ".eps".

Definition rules for reg=, and reginbox = are the same as that for the region mesh in section 5.1.1.

For saving calculating time, an outer region defined by the radius r-out is introduced additionally. You have to use a larger r-out value when you use large geometry, or you want to put the light source and view point far away. This new definition of the outer region can be seen in input echo. Therefore, you can not use an input echo by icntl=11 as an input for next calculation.

Shadow is not created if the view point and light source are set same position.

6.17.1 box definition

Maximum 5 penetration boxes can be defined. Defined boxes become transparent. To define the box, you first set three points as $\mathbf{b}_0(x_0,y_0,z_0)$, $\mathbf{b}_1(x_1,y_1,z_1)$, and $\mathbf{b}_2(x_2,y_2,z_2)$. We define the 4-th point $\mathbf{b}_3$ from $\mathbf{b}_0$ by L cm on the vertical direction of the plane defined by these three points, i.e. ($\mathbf{b}_2$ - $\mathbf{b}_0$) direction. In this `box` definition, you can use the coordinate transformation as `trcl` = transform number or `trcl = (.....)` before the definition of the points.

The box definition is shown below. Each relations are also shown in Fig. 6.8.

```
box = 2
box   x0  y0  z0
      x1  y1  z1
      x2  y2  z2  L
box   trcl = 2
      x0  y0  z0
      x1  y1  z1
      x2  y2  z2  L
box   *trcl = (0 0 0  0 90 90  90 60 150  90 30 60  -1)
      0.0  0.0  0.0
     -5.0  0.0  0.0
      0.0  0.0  5.0   5.0
```

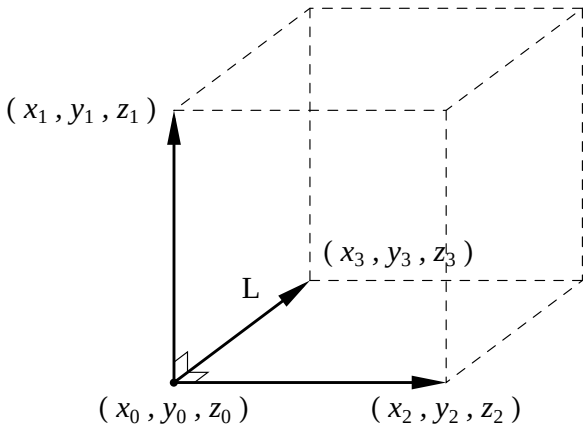


Figure 6.8: Example of box definition.

6.17.2 3dshow example

List 6.6 ● [t-3dshow] example

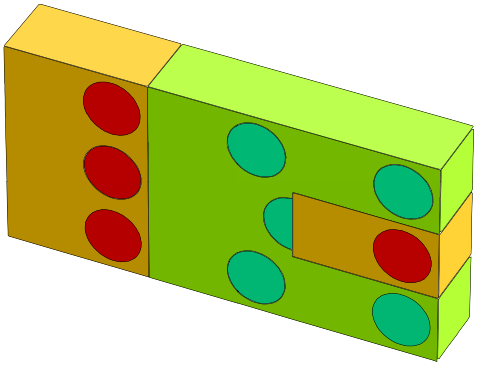
```
 1:  [cell]
 2:      1  0 -1 fill=1
 3:      2  0 -41 42 -43 44 -45 46  u=1 fill=5
 4:     22  0 -41 42 -43 44 -45 46  u=1 trcl=(0 0 20) fill=6
 5:     23  like 22 but trcl=(0 0 40) fill = 7
 6:      5  0 -21 22 -23 24 -25 26  u=5 lat=1 fill=3
 7:      6  0 -21 22 -23 24 -25 26  u=6 lat=1 fill= -1:1 0:0 0:0  2 2(0 0 5) 2
 8:      7  0 -21 22 -23 24 -25 26  u=7  fill= -1:1 0:0 0:0  2 3 2 lat=1
 9:      3  1  3.97300E-02  3 u=2
10:      4  6  4.18280E-02 -3 u=2
11:     13  5  8.47130E-04 -3 u=3
12:     14  3  1.23620E-01  3 u=3
13:     8  -1 +1
14:  [surface]
15:      1   rpp -15 15 -5 5  -5 55
16:      21   px    5
17:      22   px   -5
18:      23   py    5
19:      24   py   -5
20:      25   pz   15
21:      26   pz   -5
22:      41   px   15
23:      42   px  -15
24:      43   py    5
25:      44   py   -5
26:      45   pz   15
27:      46   pz   -5
28:      5   rpp -20 20 -5 5 -5 35
29:      6   rpp -20 20 -5 5 -5 15
30:      7   rpp -20 20 -5 5 35 55
31:      3   c/y   0 10  4
```

In above geometry, the whole body is rectangular solid, and it has rectangular solid lattices including cylinders inside. You can make graphical plot for the geometry by the 3dshow as,

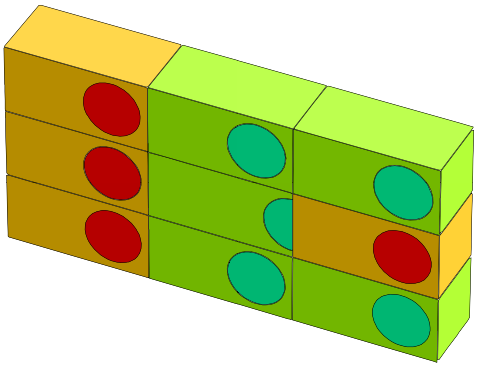
List 6.7 ● [t-3dshow] example

```
 1:  [t-3dshow]
 2:     output = 3
 3:     heaven = x
 4:     resol = 2
 5:     width = 0.1
 6:     x0 =  0
 7:     y0 =  0
 8:     z0 = 25
 9:     e-the = 70
10:     e-phi = 50
11:     e-dst = 1000
12:     l-the = 50
13:     l-phi = 25
14:     l-dst = 2000
15:     w-wdt = 60
16:     w-hgt = 40
17:     w-dst = 150
18:     file = dshow.dat
```

The output result is ,

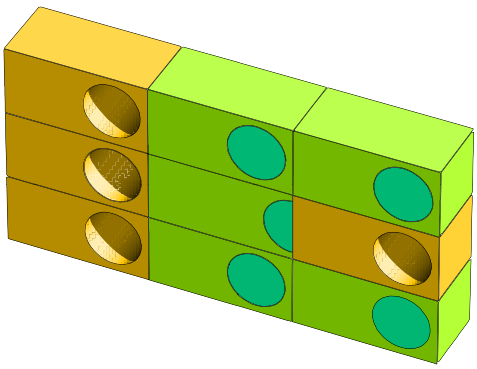


You can add region boundary by option `line=1` as,



You can see how lattices are set up. Next, let material number 5 be transparent, and add shadows by

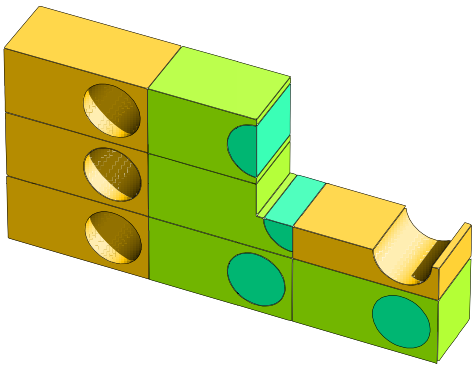
```
material = -1
            5
  shadow =  2
```



Let's define a box.

```
box = 1
box     0  10  30
      100  10  30
        0  10  100 100
```

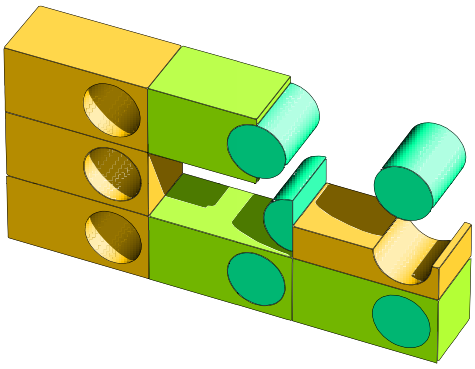
The defined box part becomes transparent, and you can see inside of the body.



In the last example, add

```
reg = ( 3 < 6[0 0 0] )
matinbox = 1
           6
```

Regions defined by `reg = (3 < 6[0 0 0] )` become transparent, and material number 6 becomes visible.



You can display any complex structures as you like combining with these options.

7 Volume and Area calculation by tally function

Sometimes, you need to obtain values of region volumes and areas of crossing surface for tally definitions. You can obtain these values by using the tally itself by Monte Carlo method.

In order to calculate volume and area by Monte Carlo method, you have to calculate the flux pass through the region or the crossing surface by making use of the spatially uniform trajectories. You can make uniform trajectories by using the cylinder (s-type=1,4) or the prism (s-type=2,5) with the disc source (z1=z0 in cylinder source) or rectangular source (z1=z0 in prism source), and with constant direction. For the volume calculation, you can use the track tally with `unit=4`, and giving 1 for volume input. For the area calculation, you can use the cross tally with `unit=1`, and giving 1 for area input. As a factor, for both cases, you should put the area of the rectangular source or the area of the disc source. The results of the flux give you the values of volume (cm^3) or area (cm^2) which you need. If you give an accurate value to the volume or area input, resulting flux must be 1 when Monte Carlo calculation is correct.

The weak point of this method is, that the error does not decrease with number of histories easily when there are structures or surfaces parallel with the beam direction. In addition, `r-in` and `r-out` definitions becomes difficult in the cross tally. In order to resolve the problem, the spherical shell source (s-type=9,10) is prepared. Set `r1=r2` in order to use the spherical shell source. And set `dir=-all` in order to make an inside direction source with `cos` distribution. Particle trajectories by the source become uniform in the sphere. In addition, this source is given a $\cos^2$ bias in order to make good statistics in center region. You have to set πr^2 as the factor for volume and area calculations. In the case you define one-way crossing surface tally for the r-in and r-out (see section 6.2), you have to set $2\pi r^2$ as the factor.

For all cases mentioned above, you should set `icntl=5` for non reaction calculation. The volume and area calculations are also useful for check of your geometry. You may find some geometry errors after the calculation.

An example using the spherical shell source is shown below.

List 7.1 ● Source example for volume, and area calculation

```
 1:   [ S o u r c e ]
 2:    s-type =   9
 3:      proj =   proton
 4:        e0 =   500.0
 5:        x0 =   0.0
 6:        y0 =   0.0
 7:        z0 =   30.0
 8:        r1 =   18
 9:        r2 =   18
10:       dir =   -all
```

In this example, a sphere with the center (0, 0, 30) and radius 18 cm, is defined. Decide the center and radius in which an interest region or crossing surface is included in the sphere. You can set any projectiles and energies.

List 7.2 ● Tally example for volume calculation

```
 1:   [ T - T r a c k ]
 2:        mesh = reg
 3:        reg = 1 2 3 4 5
 4:      e-type = 2
 5:        emin = 0.
 6:        emax = 1000.0
 7:          ne = 1
 8:        axis = reg
 9:        unit = 4
10:        file = volume.dat
11:      factor = 18**2*pi
12:        volume
13:        non    reg     vol
14:          1      1   1.0000E+00
15:          2      2   1.0000E+00
16:          3      3   1.0000E+00
17:          4      4   1.0000E+00
18:          5      5   1.0000E+00
```

A tally example for volume calculation is shown above. As the example, define an interest region, set 1 to a group energy region including the source energy, set the `unit=4`, and set the factor as πr^2. In the case the volume input is set 1 automatically because of `unit=4`, so you do not need the volume section here. You can obtain a volume value by this tally, and you can used it as input data for volume definition in your actual calculation.

List 7.3 ● Tally example for area calculation

```
 1:   [ T - C r o s s ]
 2:        mesh = reg
 3:        reg = 3
 4:        r-in  r-out   area
 5:           1      2   1.0000E+00
 6:           2      3   1.0000E+00
 7:           3      4   1.0000E+00
 8:      e-type = 2
 9:        emin = 0.
10:        emax = 1000.0
11:          ne = 1
12:        axis = reg
13:        unit = 1
14:        file = area.dat
15:      factor = 18**2*pi*2
```

A tally example for area calculation is shown above. As the example, define an interest surface, set 1 to a group energy region including the source energy, set the `unit=1`, and set the factor as $2\pi r^2$. You can obtain a area value by this tally, and you can used it as input data for area definition in your actual calculation.

When you set `r-in`, `r-out` as

```
 4:        r-in    r-out    area
 5:        ( 1 2 ) ( 1 2 )  1.0000E+00
 6:        ( 2 3 ) ( 2 3 )  1.0000E+00
 7:        ( 3 4 ) ( 3 4 )  1.0000E+00
```

In this case, you don't need factor 2, i.e., you can use πr^2 as the factor.

8 Processing dump file

You can write down the information on transport particles on dump file by [t-cross], [t-time], [t-product] tallies. If you set the dump file as a source, you can calculate the sequential transport. Furthermore, you can get the information which cannot be obtained by the tally functions in $PHITS$ by processing the dump file. To process the dump file, however, you need to make program to process the dump file. In the following, we show a program to process the dump file as an example of such program.

The following program is a simple program which converts the ascii dump file to binary dump file, and vice versa. The following simple program could help you to make a program to process the dump file. The source program dump-a.f is attached in the holder “src” and the execute file dump_a.exe in Windows system is include in the holder “bin”.

List 8.1 ● source code of dump-a.f

```
 1: ************************************************************************
 2: *                                                                      *
 3: *     This program exchanges the binary data and the ascii data        *
 4: *     of dump file.                                                    *
 5: *                                                                      *
 6: *       modified by K.Niita on 2005/08/15                              *
 7: *                                                                      *
 8: *                                                                      *
 9: *                                                                      *
10: *                                                                      *
11: ************************************************************************
12:       implicit real*8 (a-h,o-z)
13: *-----------------------------------------------------------------------
14:       dimension isdmp(0:30)
15:       dimension jsdmp(0:30)
16:       data isdmp / 31*0 /
17:       data jsdmp / 31*0 /
18:       character chin*80
19:       character chot*80
20:       logical exex
21:       character dmpc(30)*4
22:       data dmpc / '  kf','   x','   y','   z','   u','   v','   w',
23:      &            '   e','  wt','  tm','  c1','  c2','  c3',
24:      &            '  sx','  sy','  sz','  n0','  nc','  nb','  no',
25:      &            '    ','    ','    ','    ','    ','    ',
26:      &            '    ','    ','    ','    '/
27:       dimension dmpd(30)
28:       dimension dmpp(30)
29:       data dmpp /  2112.,   0.0,   0.0,   0.0,   0.0,   0.0,   1.0,
30:      &             100.,   1.0,   0.0,   0.0,   0.0,   0.0,
31:      &             0.0,    0.0,   0.0,   0.0,   1.0,   1.0,   1.0,
32:      &             0.0,    0.0,   0.0,   0.0,   0.0,   0.0,   0.0,
33:      &             0.0,    0.0,   0.0/
34: *-----------------------------------------------------------------------
35:         in = 5
36:         io = 6
37:         id = 20
38:         ia = 21
39:         iserr = 0
40: *-----------------------------------------------------------------------
41: *       user program frag : 0 => no,  1 => with user program
42: *-----------------------------------------------------------------------
43:         iuser = 0
44: *-----------------------------------------------------------------------
45: *       read ascii or binary frag
46: *-----------------------------------------------------------------------
47:         write(io,*) ' ** 0 => read binary to ascii'
48:         write(io,*) ' ** 1 => read ascii  to binary'
49:         read(in,*,end=993) iasb
50: *-----------------------------------------------------------------------
51: *       read the name of input dump file
52: *-----------------------------------------------------------------------
53:         write(io,*)
54:         write(io,*) ' ** put the file name of input dump file'
```

```
 55:           read(in,'(a80)',end=998) chin
 56:              inquire( file = chin, exist = exex )
 57:              if( exex .eqv. .false. ) then
 58:                write(io,*) ' ** Error : the file does not exist'
 59:                goto 999
 60:              end if
 61:              if( iasb .eq. 0 ) then
 62:                open(id, file = chin,
 63:      &              form='unformatted',status = 'old' )
 64:              else
 65:                open(id, file = chin,
 66:      &              form='formatted',status = 'old' )
 67:              end if
 68:  *---------------------------------------------------------------------
 69:  *        read the number of data and data sequence
 70:  *---------------------------------------------------------------------
 71:           write(io,*)
 72:           write(io,*) ' ** put the number of data in a record'
 73:           read(in,*,end=997) isdmp(0)
 74:           write(io,*)
 75:           write(io,*) ' ** put the ID numbers of data in a record'
 76:           read(in,*,end=996) ( isdmp(i), i = 1, isdmp(0) )
 77:              do k = 1, isdmp(0)
 78:                 if( isdmp(k) .gt. 20 .or.
 79:      &              isdmp(k) .le.  0 ) goto 992
 80:                jsdmp(isdmp(k)) = k
 81:              end do
 82:              write(io,*)
 83:              write(io,'('' # dump data : '',30(a4))')
 84:      &         ( dmpc(isdmp(j)), j = 1, isdmp(0) )
 85:  *---------------------------------------------------------------------
 86:  *        read the name of output dump file
 87:  *---------------------------------------------------------------------
 88:           write(io,*)
 89:           write(io,*) ' ** put the file name of output'
 90:           read(in,'(a80)',end=998) chot
 91:              inquire( file = chot, exist = exex )
 92:              if( exex .eqv. .true. ) then
 93:                write(io,*)
 94:                write(io,*) ' ** Warning : the file already exists'
 95:                write(io,*) ' ** Do you want to overwrite ?'
 96:                write(io,*) ' ** Yes <= 0, No <= 1'
 97:                read(in,*,end=995) iyes
 98:                if( iyes .ne. 0 ) goto 999
 99:              end if
100:              if( iasb .eq. 0 .or. iuser .ne. 0 ) then
101:                open(ia, file = chot,
102:      &              form='formatted',status = 'unknown' )
103:              else
104:                open(ia, file = chot,
105:      &              form='unformatted',status = 'unknown' )
106:              end if
107:  *---------------------------------------------------------------------
108:  *        read the number of records to read
109:  *---------------------------------------------------------------------
110:           write(io,*)
111:           write(io,*) ' ** put the number of records to read'
112:           write(io,*) ' ** all <= 0, or positive integer'
113:           read(in,*,end=994) irec
114:  *---------------------------------------------------------------------
115:  *        start reading the data
116:  *---------------------------------------------------------------------
117:           write(io,*)
118:           write(io,*) ' ** start read and write the data'
119:  *---------------------------------------------------------------------
120:           jrec = 0
121:    100    jrec = jrec + 1
122:           if( irec .gt. 0 .and. jrec .gt. irec ) goto 500
123:    687       continue
124:              if( iasb .eq. 0 ) then
125:                 read(id,end=688,err=690)
```

```
126:       &         ( dmpd(isdmp(k)), k = 1, isdmp(0) )
127:              else
128:                 read(id,'(30(1p1d24.15))',end=688,err=690)
129:       &         ( dmpd(isdmp(k)), k = 1, isdmp(0) )
130:              end if
131:              goto 689
132:    688       if( irec .gt. 0 ) then
133:                 rewind id
134:                 goto 687
135:              else
136:                 goto 500
137:              end if
138:    690       continue
139:              iserr = iserr + 1
140:              write(io,'('' ** Error in dump file no ='',i5)') iserr
141:              goto 687
142:    689       continue
143:   *-----------------------------------------------------------------------
144:   *     user program here
145:   *-----------------------------------------------------------------------
146:         if( iuser .ne. 0 ) then
147:              do k = 1, 20
148:                 if( jsdmp(k)  .gt. 0 ) dmpp(k) = dmpd(k)
149:              end do
150:                 kf = nint( dmpp(1) )
151:                 x  = dmpp(2)
152:                 y  = dmpp(3)
153:                 z  = dmpp(4)
154:                 u  = dmpp(5)
155:                 v  = dmpp(6)
156:                 w  = dmpp(7)
157:                 e  = dmpp(8)
158:                 wt = dmpp(9)
159:                 t  = dmpp(10)
160:                 n1 = nint( dmpp(11) )
161:                 n2 = nint( dmpp(12) )
162:                 n3 = nint( dmpp(13) )
163:                 sx = dmpp(14)
164:                 sy = dmpp(15)
165:                 sz = dmpp(16)
166:                 n0 = nint( dmpp(17) )
167:                 nc = nint( dmpp(18) )
168:                 nb = nint( dmpp(19) )
169:                 no = nint( dmpp(20) )
170:         end if
171:   *-----------------------------------------------------------------------
172:   *     write data on the file
173:   *-----------------------------------------------------------------------
174:         if( iuser .eq. 0 ) then
175:              if( iasb .eq. 0 ) then
176:                 write(ia,'(30(1p1d24.15))')
177:       &         ( dmpd(isdmp(k)), k = 1, isdmp(0) )
178:              else
179:                 write(ia)
180:       &         ( dmpd(isdmp(k)), k = 1, isdmp(0) )
181:              end if
182:         end if
183:   *-----------------------------------------------------------------------
184:            goto 100
185:   *-----------------------------------------------------------------------
186:   *        end of process
187:   *-----------------------------------------------------------------------
188:     500    continue
189:            write(io,*)  ' ** end of read and write the data'
190:            write(io,'('' ** number of processed records is '',
191:       &         i8)') jrec-1
192:            write(io,*)
193:            close( id )
194:            close( ia )
195:            goto 999
196:   *-----------------------------------------------------------------------
```

```
197:    992 continue
198:        write(io,*) ' ** Error : ID should be 1 - 20'
199:        goto 999
200:    993 continue
201:        write(io,*) ' ** Error : the ascii or binary frag is wrong'
202:        goto 999
203:    994 continue
204:        write(io,*) ' ** Error : the number of records is wrong'
205:        goto 999
206:    995 continue
207:        write(io,*) ' ** Error : the answer should be 0 or 1'
208:        goto 999
209:    996 continue
210:        write(io,*) ' ** Error : the ID numbers is wrong'
211:        goto 999
212:    997 continue
213:        write(io,*) ' ** Error : the number of data is wrong'
214:        goto 999
215:    998 continue
216:        write(io,*) ' ** Error : file name is wrong'
217:        goto 999
218:    999 continue
219:        stop
220:        end
```

The input parameters are read from normal input, i.e. from console, in an interactive way. When you execute the program, it asks you as,

```
** 0 => read binary to ascii
** 1 => read ascii  to binary
```

You put 0 for binary, 1 for ascii. Next it asks you the name of target dump file.

```
** put the file name of input dump file
```

You put the name of target dump file.

```
** put the number of data in a record
```

The program ask you the number of data in a record. You put positive number for both ascii and binary.

```
** put the ID numbers of data in a record
```

You put ID for the data. See kind of dump data and ID, in Tables 5.3, 5.4.

```
** put the file name of output
```

You put the file name of output. If the file already exists, the program asks you whether the file can be overwritten or not.

Next, the program asks you how many records are processed.

```
** put the number of records to read
** all <= 0, or positive integer
```

If this number is larger than total record number, the program turns back to the top of the data. Finally, the number of records actually processed is shown.

When you make a program based on this program, you should change `iuser` to 1 at 35 line in above list. Then the program does not write the converted data on file. In this case, the output is written by ascii.

In 150-169 lines, there are variables “kf, x, y, z, u, v, w, e, wt, t, n1, n2, n3, sx, sy, sz, n0, nc, nb, no”. Here kf means the kf-code of the particles (see Table 3.4), x, y, z are coordinate (cm), u, v, w denote the unit vector of the direction of the particle, e is the energy (MeV, or MeV/nucleon for nucleus), wt is the weight, time is the initial time (ns), c1, c2, c3 are the values of counters, and sx, sy, sz are the unit vector of the direction of spin, respectively. By using these variables, you can make a program to obtain desired quantities.

9 Output cutoff data format

The information for neutron, photon, electron, positron, and proton below the cut off energy can be written in the output file (file(12), file(13), and file(10)), in order to continue these transport calculation by other Monte Carlo codes such as MCNP and EGS4. The data are written in binary. The format is shown below.

```
rd, rn, ( data(i), i = 1, nint(abs(rd)) )
rd, rn, ( data(i), i = 1, nint(abs(rd)) )
.................
.................
```

First, in the case of `incut = 1`, and no importance option (rd<0)

```
rd, rn, x, y, z, ( e(i), u(i), v(i), w(i), i = 1, n )
rd, rn, x, y, z, ( e(i), u(i), v(i), w(i), i = 1, n )
.................
.................
```

Next, `incut =1` with importance option (rd>0),

```
rd, rn, x, y, z, ( e(i), u(i), v(i), w(i), wt(i), i = 1, n )
rd, rn, x, y, z, ( e(i), u(i), v(i), w(i), wt(i), i = 1, n )
.................
.................
```

`incut = 2` and no importance option (rd<0),

```
rd, rn, x, y, z, ( e(i), u(i), v(i), w(i), t(i), i = 1, n )
rd, rn, x, y, z, ( e(i), u(i), v(i), w(i), t(i), i = 1, n )
.................
.................
```

`incut = 2` with importance option (rd>0),

```
rd, rn, x, y, z, ( e(i), u(i), v(i), w(i), wt(i), t(i), i = 1, n )
rd, rn, x, y, z, ( e(i), u(i), v(i), w(i), wt(i), t(i), i = 1, n )
.................
.................
```

where `n = nint(rn)`, `x, y, z` is a coordinate (cm), `e(i)` is an energy (MeV), `u(i), v(i), w(i)` is an unit vector of momentum, `wt(i)` is an weight, and `t(i)` is time (ns). In the case `igcut =3`, the particle identifier `p(i)` is written instead of `t(i)` in the case of `incut =2`.

`p(i) = 3.0` is photon, `p(i) = 4.0` is electron, and `p(i) = 5.0` is positron.

10 Supplementary explanation for region error checking

When you make a complicated geometry, it is easy to mistake the region definition such as double defined, and non-defined regions. In the cases, results might have some uncertainties even if the calculation is finished normally. You can see the summary of region error in the last part of calculation summary. If some errors are found, check your geometry definition. Results are not certifiable when region errors exist. When you make the complicated geometry, you are recommended to use graphical geometry viewers such as `CGVIEW` and `MARS-PF` with `icntl = 2, 4` options. Especially `CGVIEW` can survey if region errors exist or not.

The geometry checking method only by *PHITS* is explained in followings. First, set `icntl =5` for non-reaction and non-ionization processes. Then set small value into the `deltm` as `deltm=1`. (you should add fraction in order to avoid the `deltm` becomes integer multiple of a distance between regions) `igchk =1` is required to examine region crossing particles. You can use the default value for the `deltb`. Then, set as `s-type = 1 - 5` and `dir = all` .

By the definition, an isotropic source is used. Now run *PHITS* with small number of histories to see the CPU time with these options. Then set an actual number of histories for geometry check calculation, and run the code. If some errors are found, it is output in the standard output.

In addition, if you add some value into the `igerr` as `igerr = 10` , a particle can go through error regions and look for further error regions. You may find additional error regions by the option.

11 Additional explanation for the parallel computing

There are two types of parallel computing method; one is the distributed-memory parallel computing using MPI protocol, and the other is the shared-memory parallel computing using OpenMP architecture. You can perform the parallel *PHITS* calculation using both methods. In addition, the hybrid of them can be also utilized. To execute distributed-memory parallel computing, you have to install MPI protocol in your computer. On the other hand, there is no required software for the shared-memory parallel computing. You have to make each executable file according to the type of the parallel computing. See Sec. 2.4 in detail.

In the distributed-memory parallel computing, jobs are distributed to each CPU core in unit of batch. When all jobs assigned to each core are finished, the main core gathers their results. In this mode, all cores individually use the memory equivalent to that used in single processing. Thus, the total RAM memory in the computer system must be larger than that used for single processing multiplied by the core number. Therefore, this parallel computing type is not suitable for the calculation requiring a large memory such as that using voxel phantom.

In the shared memory parallel computing, jobs are distributed to each CPU core in the unit of history. Then, all cores share a large part of memories used in *PHITS* , except for those defined as "thread private" variable. Therefore, the memory required in this parallel computing is almost the same as that in single processing. A disadvantage of this parallel computing in comparison to memory-distributed one is the slower computational time due to the competition of accessing the shared memories. This disadvantage becomes very important for calculations frequently updating memories such as those using the `[t-sed]` tally.

11.1 Distributed memory parallel computing

11.1.1 How to execute

The *PHITS* calculation using the distributed memory parallel computing with MPI-protocol can be executed by the following command;

```
mpirun -np 8 phits_lin.exe
```

where "phits_lin.exe" indicates the *PHITS* -executable file name. The number of PE (Processing Element) should be set after "-np" . You can send this command using the parallel computing submission protocol such as "qsub" . In this case, the name of the *PHITS* input file should be written in a text file named "phits.in" whose 1st line is:

```
file = input_file_name
```

where "input_file_name" is the name of the *PHITS* input file. This rule is only effective for the distributed memory parallel computing. You can also write `file=phits.in` at the 1st line, and add the contents of *PHITS* input file after the 2nd line of "phits.in" . Please see Sec. 2.4 in more detail.

11.1.2 Adjustment of maxcas and maxbch

In the distributed-memory parallel computing, jobs are distributed to each CPU core in unit of batch. Hence, the number of batch (`maxbch`) should be a multiple of PE−1 (one of PEs is used for control). If not, *PHITS* automatically changes `maxbch` to be a multiple of PE−1, and adjusts the number of history per batch `maxcas` to make the total history number equivalent to that you set in the input file. In this case, some comments are outputted at the end of the input echo.

In the case of the restart mode (`istdev<0`), adjustment of `maxcas` is not performed, since it should be set to the same as written in the past tally results.

11.1.3 Treatment of abnormal end

When *PHITS* stops by abnormal end in a PE, the PE is removed from operation. Finally, a total result by remained PE is given as a final result. In this case, you should pay attention for the `ncut` file. The `ncut` is incomplete.

11.1.4 ncut, gcut, pcut and dumpall file definition in the PHITS

For the parallel calculation, ncut, gcut, and pcut can be defined in an input file as normally as

```
file(12) = temp/ncut.dat
```

In 1 PE calculation, specified ncut.dat is written normally, but in multi PE calculation, ncut.dat is written separately in each node as

```
/wk/j9999/temp/ncut.dat
```

where "j9999" is your user-name which is read in automatically from the environmental variable LOGNAME. By default, your user-name is put in the LOGNAME in the UNIX system.

Before parallel calculation, make j9999 directory under the /wk directory for each node. If you want to make ncut file in a directory not named by your user-name, change environmental variable LOGNAME before parallel calculation. In the case, confirm there exists the directory you specified under the /wk.

inpara, igpara, and ippara are prepared for writing options. By default, they have zero value. If you give value 1, output files are given IP numbers as

```
/wk/j9999/temp/ncut.dat.005
```

where 005 is the IP numbers.

If you give 3 into inpara, igpara, and ippara, the default file path

```
/wk/j9999/
```

is not added. 3 puts IP number at the end of file as same as 1.

11.1.5 Read in file definition in the PHITS

Read in files for *PHITS* are the trxcrd.dat, the data file for photon emissions from residual nuclei, and the Decay-Turtle source file. The former is 2.6MB size file and is read only one time. It gives only small effect to network traffic. So the trxcrd.dat can be located at one place. But latter is, sometimes, 100MB size and is read in every events. It gives a large effect to network traffic. So you should copy the Decay-Turtle data file and put them in each PE as /wk/j9999/turtle/sours.dat, and define as file = /wk/j9999/turtle/sours.dat in the *PHITS* input.

11.2 Shared memory parallel computing

11.2.1 Execution

The *PHITS* calculation using the shared memory parallel computing can be executed by the following command;

```
phits_lin.exe < phits.inp
```

where "phits_lin.exe" indicates the *PHITS* -executable file compiled with OpenMP option, and "phits.inp" does the *PHITS* input file. You can use any name for the *PHITS* input file, namely the restriction of the *PHITS* input file name to "phits.in" is not valid in this case. To specify the number of cores for the parallel computing, an environment variable "OMP_NUM_THREADS" should be defined. It should be noted that this variable should be set to the real number of CPU-cores, not the number of total cores in the case of using the hyper-threading technology, since the parallel computing using this technology does not work in the *PHITS* calculation. You can change the environment variable by the following way;

```
set OMP_NUM_THREADS=4 (for Windows)
export OMP_NUM_THREADS=4 (for Mac and Linux)
```

In order to execute memory-shared parallel computing *PHITS* using "sendto" command, you have to change the environmental variable "PHITS_PARALLEL" written in "\phits\bin\phits.bat" to the number of cores that you want to use. It should be noted that in the parallel computing of hybrid type you have to set individually the environment variable on all nodes.

11.2.2 Important notices for shared memory parallel computing

If you use only 1 core for the memory-shared parallel computing *PHITS* , it takes approximately twice as much time as that in the single processing. Therefore, it is meaningless to select the memory-shared parallel computing in the case that your computer has only 1 or 2 cores.

The following functions are still in inapplicable to the shared memory parallel computing.

(1) Geometries on CG (Combination Geometry).

(2) Calculations using EGS5.

(3) Calculations of nuclear reactions using INC-ELF.

(4) Runtime execution of *ANGEL* from *PHITS* by setting `epsout=1`. (Only for Intel Fortran compiled *PHITS* on Windows.)

In order to avoid the 4th problem, the *PHITS* executable file for memory-shared parallel computing contained in the current *PHITS* package was compiled by replacing “a-angel.f” by “a-angel-winopenmp.f” , which outputs the file names specified by tallies with `epsout=1` into “angel-temporary.inp” . Then, “phits.bat” automatically executes stand-alone version of *ANGEL* “angel.bat” , employing “angel-temporary.inp” as the input file.

If segmentation errors occur in executing memory-shared parallel computing *PHITS* on Linux, it may due to the overuse of the stack memory. In that case, you have to increase the stack size using the command below:

```
export OMP_STACKSIZE=1G
```

This command sets the stack size to 1GB.

In principle, the results obtained from single processing and shared-memory parallel computing should be the same. Thus, if you found inconsistencies between the results obtained by the two modes, please let us know. It may be a bug in the programming.

12 FAQ

12.1 Questions related to parameter setting

Q1.1 How can we precisely simulate motion of low-energy neutrons? How can we use nuclear data library in *PHITS* ?

A1.1 In general, nuclear reaction models such as Intra-Nuclear Cascade (INC) or Quantum molecular dynamics (QMD) cannot simulate reaction processes attributed to induced neutrons below 20 MeV because of complex resonance structures. Therefore, nuclear data libraries are very useful for the simulation. To utilize the data library, `emin(2)` and `dmax(2)` in the `[parameters]` section have to be set appropriately, e.g. `emin(2)=1.0e-10` and `dmax(2)=20.0`.

Q1.2 How can we simulate motion of photons, electrons and positrons?

A1.2 As the default setting of *PHITS* , photons, electrons and positrons are immediately cutoff when they are created. In order to transport these particles, you have to obtain their atomic data libraries such as "jphlib01" and "jellib01" (These libraries are included in the *PHITS* package from version 2.52). In addition, you have to set `emin(12-14)` and `dmax(12-14)` in the `[Parameters]` section. The typical minimum and maximum values of these parameters are 1.0e-3 and 1.0e3 for `emin(12-14)` and `dmax(12-14)`, respectively, although `dmax(14)` can be extended up to 1.0e5. However, if you set `emin(12)` and `emin(13)` to 1.0e-3, computational time becomes very long. Thus, the recommended `emin(12)` and `emin(13)` in most cases are 1.0e-1. Please see "recommendation" in more detail.

Q1.3 What is the setting of nuclear reaction models giving the most accurate result?

A1.3 In general, the default models give the best results in most cases from version 2.50. It should be noted that you have to use nuclear data library to simulate neutrons below 20 MeV (see A1.1).

Q1.4 What kind of simulation does event-generator mode suit for?

A1.4 Event generator mode suits simulations by which the event-by-event information is necessary to be obtained, e.g. detector response calculations and design of semi-conductor devices. It is also useful for the simulation that must determine energy and type of charged particles produced by low-energy neutron interactions. In concrete, event generator mode generally suits does the simulations using `[T-Deposit]`, `[T-LET]`, `[T-SED]`, `[T-Yield]` and/or `[T-Product]` tallies. On the other hand, it is not suit for the simulations only using `[T-Track]` and/or `[T-Cross]` tallies, such as shielding calculation. See "4.2.19 Event Generator Mode" section in more detail.

Q1.5 When should I change the mode for the statistical uncertainty (the setting of `istdev`) ?

A1.5 We generally recommend the history variance mode (`istdev=abs(2)`), where the statistical uncertainty depends on the total history number (`maxcas*maxbch`), except for the case of shared-memory parallel computing, where only the batch the batch variance mode (`istdev=abs(2)`) can be selected. However, the computational time occasionally becomes extremely long in the history variance mode, especially in the case of tallies using a lot of memories, e.g. xyz mesh tally with very fine structure. When you perform the *PHITS* calculation with such conditions, please change to the batch variance mode and set `maxbch` to be more than 10.

12.2 Questions related to error occurred in compiling or executing PHITS

Q2.1 I got an error in compiling *PHITS* . How can I compile *PHITS* ?

A2.1 The status of most variables used in *PHITS* was changed from "static" to "dynamic" . Consequently, *PHITS* 2.50 or later cannot be compiled by old FORTRAN compilers such as f77 and g77. Therefore, FORTRAN compilers recommended by *PHITS* office are Intel Fortran Compiler 11.1 (or later) and gfortran 4.71 (or later). See "2.4 Compilation using makefile for Linux&Mac" section in detail.

Q2.2 Segmentation fault occurred during the execution of PHITS.

A2.2 It might be due to the overflow of the memory used in *PHITS* . In that case, you have to increase the maximum size of memory acceptable to *PHITS* . The maximum size is defined as `mdas` parameter in `param.inc` in the `src` directory. Thus, you have to increase this number, and re-compile *PHITS* . You may also have to increase `latmax` parameter if you would like to use a huge lattice structure such as voxel phantom. See Manual "2.9 Array sizes" in more detail.

Q2.3 An error occurred when I try to use `infl:` in my *PHITS* input file.
A2.3 When `infl:` command is used in your *PHITS* input file (let name `input.dat`), you have to type `file = input.dat` at the first line of `input.dat` file. Or you have to make another input file (let name `phits.in`) whose first line is `file = input.dat`, and use `phits.in` as the input file of the *PHITS* . For example

```
phits_c.exe < phits.in > output.dat
```

See Manual "2.7 Executable file" section in more detail.

Q2.4 An error occurred when I try to execute *PHITS* on Linux or Unix console, but I can execute it on Windows using the same input file.
A2.4 Many reasons are considered to cause the error, but the most probable one is the difference of "return code" used in Linux (or Unix) and Windows. If you prepare your input file in your Windows computer, and transfer to your Linux (or Unix) system using FTP software, you have to check the status of transfer mode; i.e. you have to select "ASCII mode" in your FTP software.

Q2.5 Can *PHITS* be executed on Cygwin?
A2.5 Yes, you can. You can find the Cygwin option in "makefile" of *PHITS* .

12.3 Questions related to Tally

Q3.1 What is the difference between `[t-heat]` and `[t-deposit]` tallies?
A3.1 The values calculated by `[T-Heat]` includes the deposition energy estimated using the Kerma approximation as well as the energy of cutoff neutrons and photons. On the other hand, the value calculated by `[T-Deposit]` includes only the deposition energy from charged particles due to their ionization energy loss. Thus, if you do not employ Event-Generator mode in your simulation, you have to select `[T-Heat]` tally. `[T-Deposit]` is useful for calculating deposition energy weighted by user defined function, such as $Q(L)$ relationship for calculating dose equivalent. See Manual "6.4 [T-Deposit]" section in more detail.

Q3.2 The track length or fluence of heavy ions calculated by [t-track] or [t-cross] is strange.
A3.2 It might be due to the miss-define of the energy mesh in the tally section. The energy of heavy ions should be defined in MeV in the tally section, although it should be written in MeV/nucleon in the `[Parameters]` section.

Q3.3 Results obtained by `[T-LET]` and/or `[T-SED]` tally are strange.
A3.3 You have to check the density of material selected by `letmat`. If you select the material that is not used in your geometry, you have to define its absolute atomic densities (1H 6.893407e-2 16O 3.446704e-2) in `[Material]` section.

Q3.4 How can we estimate the statistical uncertainty from the tally output?
A3.4 Version 2.50 or later, the standard deviations or standard errors are correctly outputted in the tally results. See "4.2.2 Number of history and bank" section in detail.

Q3.5 Can I use "dump" function when I execute *PHITS* in the distributed memory parallel computing?
A3.5 From version 2.30, it works. Please ask *PHITS* office about its detailed usage.

12.4 Questions related to source generation

Q4.1 How can we normalize *PHITS* outputs when I use the isotropic source (`s-type=9` or `10`, `dir = -all`)?
A4.1 If there is nothing inside the sphere of the isotropic source, the fluence inside the sphere is normalized to $1/\pi/r_1^2$ (/source), where r_1 is the radius of the sphere. Thus, if you would like to convert the tally output (/source) to the unit fluence, you have to multiply the result with πr_1^2. It should be noted that the weight control method is employed in generating the isotropic source, and thus, the event-by-event information cannot be derived from the simulation using the isotropic source. If you would like to obtain event-by-event information for isotropic irradiation, you have to set `dir = iso` in the `[source]` section. You have to update *PHITS* if you want to use this function.

Q4.2 Source particle does not created in the cell where it should be.
A4.2 If you set the source generating surface (or point) exactly on the surface of a certain cell, *PHITS* sometimes miss-identify the cell where it should be. In this case, please move the source surface a little bit different from the cell surface.

13 Concluding remarks

We have developed multi-purpose particle and heavy ion transport Monte Carlo code system *PHITS* to simulate the particle induced nuclear reactions up to 200GeV and the nucleus-nucleus collisions. The event generator mode, which was recently incorporated in *PHITS* , can calculate, as an example, LET distributions of particles in matter or energy-deposition distributions event by event, and correlations between energy depositions in different regions on a μ-scale. This mode opens a various possibility to calculate the effects of particle radiation on biological and non-biological materials, e.g. risk for single event upsets in electronic devices.

Newly introduced tally: [t-sed] makes possible to estimate RBE of charged particles by considering the productions of δ-rays and Auger electrons based on track structure simulation. This method gives to users in various research fields a hint beyond hierarchy from microscopic to macroscopic structures. Usually it is very difficult to connect simulation codes in the different structure levels, because of huge CPU time. However, this method overcomes the difficulty by making functions fitting the results of a simulation code and incorporating another code.

PHITS has been developed under the collaboration of JAEA, RIST and KEK. The code is to be further improved in future under collaboration of these institutes together with other universities/institutes all over the world. The following items are considered to be implemented in near future:

(1) Microscopic treatment of ionization process without using the continuous slowing down approximation

(2) High-energy photo-nuclear reaction model applicable to energies above 20MeV

(3) Transports of high-energy electrons, positrons and photons by incorporating the EGS5 code[11)]

Request for future improvements as well as report of bugs and defects are very much welcome. More information on *PHITS* is available from its web site: `http://phits.jaea.go.jp`.

Acknowledgement

We would like to thank the developers of NMTC/JAM, Hiroshi Takada, Shin-ichro Meigo, Makoto Teshigawara, Fujio Maekawa, Masahide Harada and Yujiro Ikeda for their collaborations.

References

1) H. Iwase, K. Niita, T. Nakamura : “Development of general-purpose particle and heavy ion transport Monte Carlo code”, J. Nucl. Sci. and Technol. 39, 1142 (2002).

2) K. Niita, T. Sato, H. Iwase, H. Nose, H. Nakashima, L. Sihver : “PHITS- a particle and heavy ion transport code system”, Radiation Measurements 41, 1080 (2006).

3) K. Niita, et al. : “High Energy Particle Transport Code NMTC/JAM”, JAERI-Data/Code 2001-007, Japan Atomic Energy Research Institute (2001).

4) Y. Nara, N. Otuka, A. Ohnishi, K. Niita, S. Chiba : “Relativistic nuclear collisions at 10A GeV energies from p+Be to Au+Au with the hadronic cascade model”, Physical Review C61, 024901 (1999).

5) K. Niita, S. Chiba, T. Maruyama, H. Takada, T. Fukahori, Y. Nakahara and A.Iwamoto : “Analysis of the (N,xN’) reactions by quantum molecular dynamics plus statistical decay model”, Physical Review, C 52, 2620 (1995).

6) J. J. Butts and Robert Katz, “Theory of RBE for Heavy Ion Bombardment of Dry Enzymes and Viruses”, Radiation Research 30, 855-871 (1967).

7) Y. Sakamoto and Y. Yamaguchi : ”Dose Conversion Coeffcients in the Shielding Design Calculation for High Energy Proton Accelerator Facilities” JAERI-Tech 2001-042, (2001).

8) T. Sato, R. Watanabe and K. Niita, “Development of a calculation method for estimating the specific energy distribution in complex radiation fields”, Radiat. Prot. Dosim. 122, 41-45 (2006).

9) T. Sato, Y. Kase, R. Watanabe, K. Niita and L. Sihver, “Biological dose estimation for charged-particle therapy using an improved PHITS code coupled with a microdosimetric kinetic model”, Radiat. Res. 171, 107-117 (2009).

10) International Commission on Radiation Units and Measurements, Microdosimetry (ICRU report;36). Maryland: ICRU, 1983.

11) H. Hirayama et al : “The EGS5 code system”, SLAC-R-730 (2005) and KEK Report 2005-8 (2005).

Index

PHITS *Ver. 2.52 User's Manual*

Japanese edition

Ver.1.00	2002/08/12
Ver.1.20	2002/11/22
Ver.1.30	2003/02/05
Ver.1.40	2003/04/21
Ver.1.50	2003/09/09
Ver.1.62	2003/11/06
Ver.1.70	2004/01/29
Ver.1.80	2004/10/28
Ver.2.04	2005/10/27
Ver.2.05	2005/11/10
Ver.2.06	2006/01/06
Ver.2.08	2006/01/17
Ver.2.30	2011/08/18
Ver.2.30a	2011/12/02
Ver.2.30b	2012/08/23
Ver.2.50	2012/09/25
Ver.2.51	2012/10/03
Ver.2.52	2012/12/27

English edition

Ver.1.20	2003/04/23
Ver.1.70	2004/02/05
Ver.2.04	2005/10/27
Ver.2.05	2005/11/10
Ver.2.06	2006/01/06
Ver.2.08	2006/01/17
Ver.2.30	2011/08/18
Ver.2.30a	2011/12/02
Ver.2.30b	2012/08/23
Ver.2.52	2012/12/27

PHITS development members :
Koji Niita[1], Norihiro Matsuda[2], Shintaro Hashimoto[2,†], Yosuke Iwamoto[2], Hiroshi Iwase[3], Tatsuhiko Sato[2], Shusaku Noda[2], Tatsuhiko Ogawa[2], Hiroshi Nakashima[2], Tokio Fukahori[2], Keisuke Okumura[2], Tetsuya Kai[2], Takuya Furuta[4], Satoshi Chiba[5,2], and Lembit Sihver[6]

[1]Research Organization for Information Science & Technology (RIST)
[2]Japan Atomic Energy Agency (JAEA)
[3]High Energy Accelerator Research Organization (KEK)
[4]RIKEN
[5]Tokyo Institute of Technology (TITech)
[6]Chalmers University, Sweden
[†]Editor of this user's manual from ver. 2.30
