PHITS

Ver. 3.17

User's Manual

English version

Preface

This manual is the Particle and Heavy Ion Transport code System (PHITS) user's guide. PHITS is a general-purpose Monte Carlo particle transport simulation code that is used in many studies in the fields of accelerator technology, radiotherapy, space radiation, etc. For details on the physical models and important functions implemented in PHITS, see the main article [1], benchmark study [2], and papers citing them. This manual explains how to execute PHITS and which parameters should be used in the system.

The contents of this manual correspond to the PHITS version number shown on the title page and are subject to change without notice. If you have any question or comment regarding this manual, please contact the PHITS office (phits-office@jaea.go.jp).

[1] T. Sato, Y. Iwamoto, S. Hashimoto, T. Ogawa, T. Furuta, S. Abe, T. Kai, P.-E. Tsai, N. Matsuda, H. Iwase, H. Shigyo, L. Sihver, and K. Niita, Features of Particle and Heavy Ion Transport code System (PHITS) version 3.02, J. Nucl. Sci. Technol. 55, 684-690 (2018).

[2] Y. Iwamoto, T. Sato, S. Hashimoto, T. Ogawa, T. Furuta, S. Abe, T. Kai, N. Matsuda, R. Hosoyamada, and K. Niita, Benchmark study of the recent version of the PHITS code, J. Nucl. Sci. Technol. 54:5, 617-635 (2017).

Contents

1 Recent Improvements and Development Members

1.1 Recent Improvements

Essential improvements after version 2.24 are described below.

From ver. 3.17, the following function has been implemented, and some bugs have been fixed. (2019/10/29)

- A new option `foamout=2` to output the tetrahedral-mesh tally results in csv format has been implemented.
- The bug in the `[t-deposit]` calculation using EGS5 was fixed. This bug was due to uninitialization of a certain variable in PHITS, so it occasionally causes a serious influences on the tally results, but in most case not.
- The calculation procedures for 'sum over' in many tallies have been revised.
- The bug in the motion of cut-off particles with energy struggling when `nedisp=1` has been fixed.
- The parameters used in the biological dose calculation using "usrdfn2.f" were changed from those given in T.Sato `et al.` Radiat. Prot. Dosim. 143, 491-496 (2010) to T.Sato `et al.` Radiat. Res. 171, 107-117 (2009).

From ver. 3.16, the bug for `sangel` parameter has fixed. The other bug in ver. 3.15 that `[t-track]` tally does not output "`_err`" files has also been fixed. (2019/09/26)

From ver. 3.15, the following bugs have been fixed. (2019/09/12)

- A bug that `[t-deposit]` overestimates contributions of the energy cut-off in EGS5 has been fixed. Furthermore, another bug when using EGS5 and lattice structure has been fixed. The former and latter bugs occurred from version 3.13 and 3.10, respectively.
- A bug regarding the function when setting `itall=3` has been fixed. Segmentation errors occurred depending on input files. This bug occurred from version 3.13.
- A bug of DCHAIN, which occurred only in version 3.14, has been fixed.

From ver. 3.14, the following functions have been implemented, and some bugs have been fixed. (2019/08/18)

- DCHAIN-SP has been improved in terms of the following two features; (1) several neutron activation libraries and decay-data libraries have been developed based on the latest evaluated data, and (2) statistical uncertainties of the induced activities can be evaluated considering those of the production yields calculated by PHITS. Owing to these improvements, some parameters have been added to `[t-dchain]` section (see section Sec. 7.14 in detail). The new version of DCHAIN will be called DCHAIN-PHITS to be distinguished with the previous versions.
- A new function to calculate the DNA damage yields has been developed, using the track structure mode and user-defined tally. Their sample data are contained in "/phits/utility/usrtally/DNAdamage."
- Two bugs occurred only in version 3.13 have been fixed, which are related to `[t-interact]` and the track structure mode.

From ver. 3.13, the following functions have been implemented, and some bugs have been fixed. (2019/08/02)

- The default value of the cut-off energy for proton, pion, muon, and ions has been decreased from 1 MeV/u to 1 keV/u.
- A new source type (`s-type=26`) has been introduced to define a surface source which generates source particles on the surface(s) that has/have been pre-defined in the `[surface]` section. Please see Section 5.3.16 for more details.
- A new option, `itall=3`, has been implemented to plot the trend of tally results and their statistical uncertainties for each batch with the ongoing PHITS calculation. This option is available only for `[t-track]` and `[t-point]` at this moment.

- A new parameter, `istdcut`, has been introduced to the `[parameter]` section. It enables the termination of the tally calculations of which the statistical uncertainties have reached the `stdcut` setting. This function is particularly useful when the types of tallies which consume much longer computational time, such as `[t-point]` and `[t-sed]`, are written in the same input file with other tallies. Another new parameter, `istdbat`, has been introduced to inactivate `stdcut` at the early stage of the PHITS simulation.
- In stopping power calculations using ATIMA, the ionization potential of water is expected to automatically adjust to 75 eV. However, this function was valid only when the material of water was explicitly specified by the isotope 1H, i.e. material consists of natural abundant hydrogen is not treated as water in this particular respect. We modified the program, so that the material of water which is defined by naturally-abundant hydrogen isotopes, such as in the format of `H 2 O 1` and `H -0.1111 O -0.8889`, is now regard as water for stopping power calculations. Owing to this modification, the ranges of charged particles in water increase slightly (less than 1% in most cases) compared to the results obtained from the previous versions. The valid condition of `ih2o` parameter has been also revised in the same way.
- Some bugs have been fixed, such as ignorance of `stdcut` in the MPI version, strange behavior of particles when mirror surfaces are used in lattice structure, and a rare bug in EGS5.

From ver. 3.12, the following functions have been implemented, and some bugs have been fixed. (2019/06/20)

- The new parameters `cnt(1)`, `cnt(2)`, and `cnt(3)` have been introduced in `[source]` section to specify the initial counter values for each multi-source subsection <`source`>. The maximum number of <`source`> subsections is extended from 200 to 500.
- A function to consider the decrease of target radioactive nuclides due to nuclear reactions has been implemented in `[t-dchain]`. DCHAIN has also been revised in correspondance to the `[t-dchain]` improvement. Note that former versions of DCHAIN cannot read the outputs from the improved [t-dchain].
- The `itall = 1` option has become available in the Windows OpenMP version of PHITS.
- Some bugs regarding the FLUENT-PHITS coupled calculations have been fixed, including the incorrect translation of the material densities between FLUENT and PHITS.

From ver. 3.11, the following bugs have been fixed. (2019/05/16)

- The bug of MPI parallel calculations with tetrahedral-geometry stopped abruptly in ver. 3.10 has been fixed.
- The bug that the magnetic field maps given in `r-z` grid (`type = -2 or -4`) cannot be properly considered has been fixed.
- Annihilation photons are double counted in the RI source generation function when the production of both positron and photon are considered. To avoid the problem, a new parameter `iannih` has been introduced to exclude the annihilation photons in the case of `proj = photon`.

From ver. 3.10, the following functions were implemented, and some bugs were fixed. (2019/03/13)

- Procedure for coupling PHITS with thermal analysis software such as ANSYS Fluent is established. For this purpose, a function to read tetrahedral-mesh geometry written in bulk-data format of NASTRAN is implemented. A new mesh option, `mesh=tet`, is introduced in `[t-track], [t-deposit], [t-yield]`, `[t-product]`, and `[t-dpa]` to output the tallied quantities in each tetrahedral mesh. Field-data format used in OpenFoam, which can be directly read by thermal analysis software, has become selected as an output format.
- As a part of the improvements for coupling with thermal analysis software, a new option, `unit=5`, is introduced in `[t-deposit]` to output the deposition energy in J/m^3/source.
- Sample files for using high-energy nuclear data library JENDL-4.0/HE are provided in the PHITS package. Please see `\phits\recommendation\jendlHE` in more details.
- A function to read magnetic field maps written in xyz or r-z grid is implemented.
- `[t-dchain]` has become applicable to `mesh=xyz`. In reply to this improvement, DCHAIN is also updated. Note that the function to visualize the induced activities in `xyz` mesh has not been developed yet.

- Error and warning ID numbers are introduced for some messages.
Please see `\phits\manual\supplementE-ErrID.pdf` in more detail.

- A nuclear reaction model to handle the reactions by electron-, mu-, tau-neutrino and their antiparticles up to 150 MeV of incident energy is incorporated. Neutrinos can interact with orbital electrons, ^{1}H, and ^{2}H whereas the nuclei heavier than ^{2}H are currently out of the range of this model.

- INC-ELF, an intra-nuclear cascade model developed by Kyushu University, is updated to consider collective excitation of the target nucleus and to precisely calculate charged particle emission barriers.

- Electron track-structure mode is improved to consider the production of Auger electrons.

- User-defined energy resolution function has been implemented to `[T-deposit]` in order to reproduce energy resolution which cannot be realized by `dresol` and `dfano`. Users can define the functional forms of energy resolution in "usresol.f," which is activated by defining negative `dresol`. As a sample, the default "usresol.f" contains a numerical model proposed by Meleshenkovskii et al. to reproduce asymmetric peaks of a CdZnTe detector.

- A new option, `gshow=5` and `[t-gshow]` with `output=10`, is introduced to visualize the geometry in the pixel format. This option is useful for properly visualize the geometries with fine structure such as voxel and tetrahedral phantoms.

- Time for reading lattice structure is reduced by introducing compressed format

- Some parameters in the muonic atom cascade program have become adjustable.

- The third user-defined function, "usrdfn3.f," is introduced for weighting the results of `[t-deposit]`. The default program of usrdfn3.f can convert the absorbed dose to the water equivalent dose, which is frequently used in medical physics.

- Remaining batch number has become adjustable using "batch.out."

- In `[t-cross]` with `mesh=xyz` or `r-z`, a new option, `enclose=1`, is introduced for calculating incoming and outgoing particle fluxes (or currents) from each mesh.

- Deposition energies in the cells with the same ID but placed in different lattice coordinates can be separately analyzed in the case of `[t-deposit]` with `output=deposit`.

- Nuclear data libraries containing more than 1000 γ-ray spectra such as the latest version of ENDF and JEFF have become acceptable in PHITS.

- The fixed charge mode, `ifixchg=1`, is introduced for calculating the stopping power of particles with a certain charge state. Note the charge exchange reaction has not been implemented yet, so the fixed charge mode is appropriate only when target material is extremely thin at this moment.

- Several bugs are fixed, including the problem in positron transport when `emin(12)` is specified without specifying `emin(13)`, and that in electron and positron transports in the magnetic field defined at a lattice structure.

From ver. 3.08, the following changes have been made. (2018/08/20)

- The user-defined activation cross section data can be used in the `[t-yield]` and `[t-dchain]` sections. Please refer to Section 7.7 for further detail. This new function was supported by the R&D program entitled "Reduction and Resource Recycling of High-level Radioactive Wastes through Nuclear Transmutation" in the Impulsing Paradigm Change through Disruptive Technologies Program (ImPACT).

- For RI sources, characteristics x-ray and internal conversion electron are now considered.

- Tetrahedral geometries that are converted from CAD can now be used, even when the surface of tetrahedron does not perfectly match the boundary surfaces.

- The parameters (`c1-c99`) specified in the input file can be used in the user-defined source file ("usrsors.f").

- The high-speed mode of ANGEL, which does not check numerical data while plotting the results, has become the default.

- For cylindrical sources (s-type=1), a few bugs, such as the bug occurring when a source particle is sampled between 0 and r1 with the setting of r0=r1, have been fixed.

From ver. 3.07, mesh=xyz became available in [weight window] for EGS5. Conversion coefficients for soft error rates on semiconductor devices were added to default data of [multiplier]. In addition, some bugs for JAM were fixed. (2018/07/05)

From ver. 3.06, the following functions were implemented, and some bugs were fixed. (2018/05/29)

- By defining igamma as negative, Doppler effect owing to the motion of the emitter nuclei is disregarded. Specific gamma-rays are observed as mono-energetic peak in the energy spectrum.

- GEM Ver.2 including nucleon-gamma emission competition was developed. This version is available by setting ngem=2 in [parameters] section.

- New format of [transform] was implemented to define rotation around z,y,x axes simply.

- The distributed-memory parallel computing using a MPI protocol became available on Windows. This development was performed under support of NAIS Inc.

From ver. 3.05, the following functions were implemented, and some bugs were fixed. (2018/03/14)

- The function to calculate the deposition energy using kerma approximation, which used to be implemented only in [t-heat], became available in [t-deposit]. Necessity of the use of kerma approximation in the energy deposition calculation is automatically judged based on the status of e-mode and negs specified in [parameters] section. Owing to this improvement, we recommend to use [t-deposit] instead of [t-heat] in all situations of deposition energy calculations.

- [t-deposit] with output=deposit can be used in the case that weight is not always 1 unless weights of energy depositing particles change within one history. Then, this mode works in the cases that you set s-type=9 and dir=-all in [source] section, or you use [forced collisions] for neutral particles.

- The graph style for [t-cross] with axis=z is changed from histogram to dot & line.

- Specification of z-type=1 and nz=0 is allowed for [t-cross] to calculate fluences of particles passing through a certain surface.

- Numbers of delta-rays produced by ions (including protons) and knock-on electrons produced by electron scattering can be separately calculated in [t-interact] section. In addition, they can be distinguished in [counter] section too.

- RI source function was improved to produce Auger electrons even when the Auger electron spectrum data are not included in RIsource.ack.

- A bug in [t-product] in the case of iMeVperu=1 was fixed.

From version 3.04, the probability density of the number of interactions by individual history can be outputted from [t-star]. The number of interactions occurred in the track-structure mode can be also counted in the tally. In addition, the tally name was changed to [t-interact] in order to explicitly state that it is the tally for counting the number of interactions, though the former name [t-star] can still be accepted. (2018/02/16)

From ver. 3.03, the following functions were implemented, and some bugs were fixed. (2018/02/01)

- A new function to generate source particles from tetrahedron geometry was implemented. See Section 5.3.15 in more detail.

- mesh = xyz became available in [weight window] and [t-wwg] sections.

- Coefficients depending on particle type can be defined in [multiplier] section, using part parameter. Owing to this improvement, several types of radiation doses can be directly calculated using pre-defined [multiplier] sections.

- [t-deposit] with output=deposit can be used in the case that source weight is not 1.
- A bug in [t-deposit] in the cases of output=deposit, dresol$\neq$ 0, and part$\neq$all was fixed.
- A bug in the range calculation of charged nuclei in some tallies when iMeVperu=1 was fixed.

From ver. 3.02, the following functions were implemented, and some bugs were fixed. (2017/12/01)

- New parameters nudtvar and udtvar(i) were introduced for [t-userdefined]. After variables udtvar(i), ($i = 1, \cdots$, nudtvar) was set in an input file, these can be used in subroutine usrtally. There is no upper limit of the number of udtvar(i) unlike udtpara before ver. 3.01. udtpara can be also set after this version.
- New options were implemented for the unit of [t-let] and [t-sed].
- Position of the source generation was adjusted to just on the spherical surface for s-type=9 with dir=iso. This revision influences only when some materials are placed outside the source sphere.
- A bug related to [counter] section was revised. Before this revision, occurrence of atomic interactions were ignored in calculating counter when negs was set to -1 (only photon transport mode).
- A bug in the calculation of the restricted stopping power when δ-rays are generated by [delta ray] was fixed.
- The bug in the calculation of angular straggling using nspred = 2 was fixed. This bug was introduced in PHITS2.96, and calculation results for charged particle beam using the versions between 2.96 and 3.01 might be strange.

From ver. 3.01, the following functions were implemented, and some bugs were fixed. (2017/10/31)

- A special ANGEL parameter sangel was introduced to insert all ANGEL parameters into tally output files. This function was developed by Mr. Takamitsu Miura of RIST, and was supported by Center for Computational Science & e-Systems, JAEA.
- Reading algorithm for tetrahedral geometry was revised to reduce the computational time.
- ANGEL was revised to reduce the computational time for making eps files of tally results with 2-dimensional type such as axis=xy.
- A bug for ignoring stdcut in [t-deposit] was fixed.

From ver. 3.00, the Kurotama model was used as the default model to give nucleus-nucleus reaction cross sections, and the natural isotope expansion was effective in input files of DCHAIN generated by [t-dchain]. In addition, some bugs were fixed. (2017/10/04)

From ver. 2.97, the following functions were implemented, and some bugs were fixed. (2017/09/21)

- A new parameter iMeVperu was introduced to convert the unit of nucleus energy [MeV] to [MeV/u] in outputs of all tallies. By setting of iMeVperu=1 in [parameters], all tally results are output with the unit of MeV/u for the nucleus energy. This function was developed by Mr. Takamitsu Miura of RIST, and was supported by Center for Computational Science & e-Systems, JAEA.
- New parameters r-from, r-to in [t-cross] with mesh=reg were added, because conventional parameters r-in, r-out were confusing. r-from, r-to can be used instead of r-in, r-out, respectively.

- The option of the stopping power ndedx=3 became applicable for target nuclei with the mass number of $93 \leq Z \leq 97$ (from Np to Bk). See Section 5.2.8 for more detail.
- By adding "$OMP=*N*" (*N* is the number of CPU cores to be used) before the first section in a PHITS input file, the shared-memory parallel computing using OpenMP is executed. Even when infl: is set, a line of "file=*input file name*" is not required. In Mac OS, a terminal window is opened when PHITS is executed by drag and drop on the Dock. Note that these function are not available when to run PHITS on the command line on Linux and so on.

- The default value of the switching energy from JQMD to JAMQMD, `ejamqmd`, was changed to 3GeV.

From ver. 2.96, the following functions were implemented, and some bugs were fixed. (2017/08/28)

- A new section `[ww bias]` and a new tally `[t-wwbg]` were implemented to bias `[weight window]` to obtain better statistics for a certain direction. See Section 5.20 and 7.16 in more detail.
- The numerical data of tally results are outputted after each batch is finished even for `itall=0` (default). Thus, the difference between `itall=1` and 0 is whether the image (*.eps) files are generated or not. Owing to this revision, the parameter `stdcut` works in the default setting.
- Only related tallies to the setting of the `icntl` parameter work. For example, all tallies except for `[t-volume]` are disabled when `icntl=14`, while `[t-volume]` is disabled when `icntl`≠`14`. Warning is outputted when `infl` or `set` command is written in the disabled section.
- Some ANGEL parameters were introduced to change the length or time axis of figures. For example, you can draw a figure of spatial dose distribution with the axis in nm scale by setting `angel=cmnm` in the tally. See section 6.7.10 in more detail.
- PHITS execution is terminated when two or more `[parameters]` sections exist in one input file.
- Some bugs are fixed in terms of the systematic equation for calculating the giant-dipole resonance cross sections, the angular straggling function for very short range particle, high-energy nucleon-nucleon interaction model JAMQMD2, and the formula for calculating the source spectrum with the Maxwell distribution.

From ver. 2.95, the procedure for `s-type` depending on the setting of the energy of source particles was changed. The mono-energetic and energy-distributed sources can be defined by setting `e0` and `e-type`, respectively, irrespective of the value of `s-type`. If both `e0` and `e-type` are defined, the energy is decided according to the previous procedure of `s-type`. In addition, some bugs were fixed. (2017/07/21)

From ver. 2.94, bug in `[t-track]` and `[t-cross]` was fixed. Before this revision, the energy of electrons and positrons passing through vacuum is slightly different from the real value when EGS5 is used. Note that this revision only influences the tally results, and has no relation with particle transport simulation itself. In addition, bugs in track-structure mode were fixed. (2017/06/30)

From version 2.93, the following functions were implemented, and some bugs were fixed. (2017/06/16)

- A new parameter `file(1)` was introduced to specify the PHITS installation folder name. When you properly set this parameter, you do not have to specify the name of other input files i.e. `file(7, 20, 21, 24, and 25)` unless you have changed the folder structure of PHITS.
- A new parameter `nucdata` was introduced to automatically adjust `emin(2)` and `dmax(2)` parameters suitable for JENDL-4.0. The default value of this parameter is set to 1, i.e. neutrons below 20 MeV are automatically transported using nuclear data library.
- A new option for `negs` parameter, -1, was introduced. When you set `negs`=−1, PHITS treats photon transport using the original algorithm (i.e. not EGS5), and ignores the electron and positron transport. This option is selected as the default setting.
- The default value of `ides` parameter was changed to 1, i.e. photon does not produce electrons in the PHITS original algorithm. If you would like to transport electrons and positrons, you have to use EGS5.
- The default value of `igamma` parameter was changed to 2, i.e. γ-rays are produced in the de-excitation process based on EBITEM model in the default setting.
- RI source function was improved to be applicable to α and β decays including Auger electron production. See Table 5.57 in more detail.
- Track-structure mode was developed in PHITS. Using this mode, PHITS can analyze ionization, excitation, and oscillation induced by electrons and positrons event-by-event. See Section 5.13 in more detail.

- A new method for calculating the energy loss of charged particles with explicitly generating δ-rays was developed, based on their restricted stopping power. This method is selected as the default setting, i.e. `irlet=1`.
- JAMQMD, which is used for simulating nucleus-nucleus interactions above 3 GeV/u, was improved to consider the relativistic effect.
- `[t-dpa]` was improved to be capable of calculating DPA by electrons, positrons, pions etc.
- Bug in the normalization process of the self-fission source (`ispfs` option) was fixed.

From ver. 2.92, the default setting of `output` in `[t-dchain]` tally changed to `cutoff` in order to score particles stopped in specified regions. Note that in the previous setting `output=product` heavy ions produced in a thin target were scored even though the ions don't stop in the target. In addition, we replaced a place where the version of PHITS is shown in eps files by that of ANGEL. We fixed a bug about triangle prism shape source. (2017/04/18)

From version 2.91, a following function was implemented, and some bugs were fixed. (2017/02/20)

- A new function to display error bars of statistical uncertainties in eps files of tally results was implemented. This function is available by setting `epsout=2` in each tally section. It should be noted that the setting is ignored when the output format is 2-dimensional type such as `axis=xy`.
- A bug in the combination of EGS5 and `[t-cross]` was fixed. This bug affected calculations in which electrons or positrons were scored with `[t-cross]` using EGS5.

From version 2.90, following functions were implemented. (2017/02/09)

- A new function to analyze the motion of electrons and positrons in the electro-magnetic fields was implemented. This implementation was performed under support of NAIS Inc.
- The default value of the `ascat2` parameter introduced in version 2.77 was revised from 0.088 to 0.038, in accordance with the original paper [3]. The PHITS results obtained by setting `nspred = 2` without specifying `ascat2` will be changed.
- A new function to generate xyz-mesh distribution source was implemented. Using this function, you can reproduce sources having a complex spatial distribution. See Section 5.3.14 in more detail.
- A new tally named `[t-volume]` was developed in order to automatically calculate the volume of each cell. See Sections 7.17 in more detail.
- A new parameter `timeout` was introduced in the `[parameters]` section. When CPU time exceeds this value, the PHITS simulation is automatically stopped. This check function works at the end of each batch.
- A new parameter `stdcut` was introduced in each tally. When the all statistical uncertainties of the tally results become less than this value, the PHITS simulation is automatically stopped. This check function works at the end of each batch.
- Nuclear and atomic interactions are explicitly distinguished in `[t-product], [t-star]`, and `[counter]`. Information on further detailed channels such as the production of bremsstrahlung can be also deduced. See Sections 7.8, 7.13, 5.26 for more detail.

From version 2.89, following functions were implemented. (2017/01/11)

- In the default setting, `c` became unusable as comment marks in `[material]` section in order to avoid an error that the elemental symbol for carbon, C, is read as comment marks. When you use `c` as comment marks in the section, set `icommat=1` in `[parameters]` section.
- A new option of `[t-deposit]` was developed to sum up the deposit energies weighted by user defined conditions. This option can be applied to simulation, for example soft errors in semiconductor devices.

[3] G.R. Lynch and O.I. Dahl, Nucl. Instrum. Methods Phys. Res, B 58, 6-10 (1991).

- A new function to use results of tallies as an energy distribution of source particles was added in `[source]`. This function can be used by specifying `e-type=20`.
- User defined function 2 (`usrdfn2.f`) in `[t-deposit]` was changed to the new option to estimate biological dose on the basis of Microdosimetric Kinetic Model. See the paper [4] in more detail.

From version 2.88, the following functions were implemented. (2016/09/29)

- The sumtally function became applicable to the `[t-dchain]` tally.
- Two user-defined cross section function options were developed: a function to extrapolate from given data for incident energies, emission angles, and emission energies; and a function that is effective in cases in which no differential cross section data are available. Nuclear reaction models can only be used to simulate nuclear reaction events with total reaction cross section data.
- A bug occurring from ver. 2.83 in which all neutrons with energy lower than `emin(2)` decay was fixed.

From version 2.87, the following functions were implemented. (2016/09/15)

- The arrows to indicate the xyz coordinates are depicted in the `[t-3dshow]` tally.
- The algorithm for calculating tetrahedral geometry was revised to reduce the computational time.
- The geometry-error information file name format ("*.err") was changed to ("* geo.out").
- A bug in the use of `MTn`, i.e., the $S(\alpha,\beta)$ table in the `[material]` section was fixed. This bug occurred only in version 2.86.

From version 2.86, the following functions were implemented. (2016/08/23)

- A new tally `[t-wwg]` was introduced. Using this tally, it is possible to automatically determine an appropriate setting for the `[Weight Window]` section. See Section 7.15 for more detail.
- A function to output the tally results in the `xyz`-mesh in the input format of ParaView, which is an open-source, multi-platform data analysis and visualization application, was implemented: see the documents in the `/utility/ParaView/` folder for more detail. Furthermore, a function to generate a Bitmap figure of the 2-dimensional tally output was implemented. These improvements were introduced with the support of Dr. Furutaka of the Research Group for Nuclear Sensing, JAEA, and V.I.C., Inc.
- A new mode for calculating the stopping power of all charged particles using ATIMA, `ndedx=3`, was added and set as the default value.
- The name of the current batch information output file was changed from `batch.now` to `batch.out`. This file name can be specified by setting `file(22)` in the `[parameters]` section.
- The RI-source function was implemented. Using this function, PHITS can generate photon sources with the energy spectra of radioisotope (RI) decay by simply specifying the activity and name of the RIs. Nuclear decay database DECDC [5] , which is equivalent to ICRP107, was used in this function: see Table 5.57 for more detail. This improvement was performed under support of Dr. A. Endo of the Japan Atomic Energy Agency (JAEA).
- A new parameter `natural` was introduced in the `[parameters]` section. When an element is defined without specifying its mass number in the `[material]` section and `natural` is set to 1 or 2, PHITS assumes that it has natural isotope composition. Note that natural isotopes whose nuclear data are not included in JENDL-4.0 are ignored in the calculation. This improvement was performed under support of Center for Computational Science & e-Systems, JAEA.
- A new section `[Data Max]` was introduced to specify the `dmax` parameter for each nucleus and material: see Section 5.16 for more detail. This improvement was performed with the support of the Center for Computational Science & e-Systems, JAEA.

[4] T.Sato et al. Biological dose estimation for charged-particle therapy using an improved PHITS code coupled with a microdosimetric kinetic model, Radiat. Res. 171, 107-117 (2009).

[5] A. Endo, Y. Yamaguchi and K.F. Eckerman, Nuclear decay data for dosimetry calculation - Revised data of ICRP Publication 38, JAERI 1347 (2005).

- The muon nuclear reaction model was improved: see the reference [6] shown below for more detail.
- A new model for calculating deuteron-nucleus total reaction cross sections was introduced. This model can be used by setting `icrdm=1` in the `[parameters]` section. See the reference [7] shown below for more detail.
- The pion total reaction cross section model was improved and set as the default model. The improved model reproduces experimental cross section data better than the old model, which used a geometrical formula. The model used can be selected using the `icxspi` parameter.
- Several sumtally subsections can be defined in an input. Some bugs related to sumtally were fixed.

From version 2.85, the following functions were implemented. (2016/05/16)

- The high-energy heavy ion reaction model, JAMQMD, which works above 3 GeV/u, was improved to JAMQMD2 in the same manner as JQMD. The accuracy and stability of the calculation are improved, particularly for cosmic-ray simulation.
- The ATIMA algorithm for stopping power calculation in PHITS was improved. Following this improvement, PHITS simulation with high precision ATIMA is now possible in nearly the same calculation time as that of SPAR. This improvement was implemented by Mr. Akio Wada of the Research Organization for Information Science & Technology (RIST) and was supported by the Center for Computational Science & e-Systems, Japan Atomic Energy Agency (JAEA).
- The unit of the `esmin` and `esmax` parameters was changed from MeV to MeV/u. These parameters define, respectively, the minimum and maximum energies of charged particles treated in the simulation.
- A bug in the high-energy photon transport (at above approximately 10 MeV) under EGS5 was fixed.
- A bug in the capture reaction of negative muons when ^{1}H is included in the material was fixed.

Some bugs in ver. 2.84 were fixed. (2016/03/16)

From version 2.83, the following functions were implemented. (2016/03/03)

- Neutron decay can be considered with a mean neutron life time of approximately 886.7 s. Correspondingly, a very large value for `tmax` (default = 1.0e9 ns) must be used to consider neutron decay in a simulation.
- A bug in the treatment of the Doppler effect under EGS5 was fixed. Because of this bug, previous versions of PHITS scored some energies for `[t-deposit]` using `part=photon` instead of the correct value of 0.
- A bug in `[t-point]` occurring when other tallies were written with `mesh=reg` behind a `[t-point]` tally, which can corrupt the results of the other tallies, was fixed.
- A bug under which the sumtally function does not work in an input file that includes `infl` was fixed.

From version 2.82, the following functions were implemented. (2015/12/16)

- A point estimator tally `[t-point]` to calculate the particle fluence at a specific point or ring (see section 7.3 as well as `/utility/tpoint/` folder) was implemented.
- A new parameter `elastic` was added to the `[t-yield]` tally to output recoil nuclei from elastic scattering.
- A new output option `transmut` was added to the `[t-star]` tally to output star density in reactions that induce transmutation of target nuclei.
- A new option `fiss` was added in the `[counter]` section to output information on secondary particles generated through fission reactions, particularly in each generation of sequential fission.

[6] S. Abe and T. Sato, Implementation of muon interaction models in PHITS, J. Nucl. Sci. Tech. 54, 101-110 (2017).

[7] K. Minomo, K. Washiyama, and K. Ogata, J. Nucl. Sci. Tech. 54, 127-130 (2017).

- A new function was implemented in the [source] section to generate neutron sources from spontaneous fission. The multiplicity and energy spectra of neutrons under this function are taken from reference [8] shown below: see section 5.3.2 for more detail. The PHITS development team is grateful to Dr. Liem Peng Hong of NAIS, Co., Inc. for his support in developing this function.

- A new function was implemented in the [source] section to generate particles from a triangle prism: see section 5.3.13 for more detail.

- A new function was implemented in [source] section to generate particles with arbitrary time information. See section 5.3.21 in more detail.

- A new parameter NONU was added in the [parameters] section to control neutron multiplicity.

- A new function to calculate the particle fluence in sector prisms was implemented in the [t-track] tally by introducing a θ-mesh in the case of mesh=r-z.

- Some bugs in the EGS5 algorithm were fixed.

- A new function to consider the polarization of photons was implemented in the calculation of nuclear florescence resonance (NRF).

- Restart calculation using the [t-dpa] tally became feasible.

- The sumtally function became applicable to all tallies except for [t-dchain]. (From ver. 2.88, this function became applicable to all tallies.)

- The individual contribution of particles can be properly calculated using [t-deposit] with output=deposit option.

- Several bugs in the muon- and photon-induced nuclear reaction models and in JQMD-2.0 were fixed.

- Instructions on how to use tetrahedral geometry (TetraGEOM), the point estimator tally [tpoint], and the user-defined tally [usrtally] were added to the utility folder.

From version 2.81, the following functions were implemented. (2015/10/15)

- The makefile was revised to consider the dependence of each source file. Owing to this improvement, the "-j" option can be used to speed up the compilation of PHITS: please be careful to ensure that the target (executable) file name is changed in the revised makefile. This revision was performed with the support of Dr. Furutaka of the Research Group for Nuclear Sensing, JAEA.

- The limitation on the number of materials possible under EGS5 was eliminated. However, PHITS calculation may still crash owing to insufficient memory when more than a few hundred materials are defined. In addition, the maximum number of elements per material is still limited to 20.

- Bugs in [t-track] and [t-deposit] occurring under EGS5 were fixed.

- A bug in [t-dchain] was fixed to enable the proper consideration of successive lines. The maximum number of regions that can be specified in [t-dchain] was extended to 500.

- A bug in [t-deposit], mesh=reg, output=deposit using the [delta-ray] section was fixed.

- A bug in [t-heat] with mesh=r-z was fixed.

From version 2.80, the following functions were implemented. (2015/09/02)

- The function to read tetrahedral geometry (a kind of polygonal geometry) was implemented (see section 5.6.5). This implementation was carried out under the support of HUREL, Hanyang University, Korea.

- A function to produce bremsstrahlung and electron-positron pairs via muon interactions was implemented.

[8] J. M. Verbeke, C. Hagmnn, and D. Wright, "Simulation of Neutron and Gamma Ray Emission from Fission and Photofission", UCRL-AR-228518 (2014).

- A function for simulating nuclear resonance florescence (NRF) was implemented. This function enables reproduction of the excitation of nuclei and the associate production of isomers via lower energy photons. The nuclear resonance fluorescence model can be activated by setting `ipnint=2` in the `[parameters]` section.
- The sumtally function became applicable to all tallies except for `[t-dpa]` and `[t-dchain]`.
- A function to read user defined cross sections was implemented (see Section 5.17).
- The algorithm to consider energy straggling of charged particles was revised to reproduce doses around the Bragg peak more precisely.
- A new parameter `idelt` was introduced to reduce the computational time for particle transport simulation in very large gas areas. When `idelt=1`, `deltm` and `deltc` are divided by the densities of each material.
- A function to properly calculate the uncertainty of tally results was implemented for use in conjunction with the `dump` source (see Section 5.3.17).
- A new parameter `pnimul` was introduced to bias the photo-nuclear reaction cross section against the photo-atomic interaction cross section.
- A bug in the calculation of the uncertainty of `[t-yield]` was fixed.
- Several improvements related to EGS5 were made:
  - A new parameter `ipegs` was introduced to control PEGS5 execution before PHITS simulation.
  - A new parameter `imsegs` was introduced to precisely simulate multiple electron scattering upon entry of electrons into a new material. This option is original to PHITS and is not included in the original EGS5.
  - A bug in the electron transport algorithm in PHITS2.77 only (but not in PHITS 2.76 or earlier) was fixed. The bug caused PHITS2.77 to calculate an insufficiently large range of electrons.
  - The limitation on the number of materials used in PHITS, even when using EGS5, was eliminated.

From version 2.77, the following functions were implemented. (2015/05/19)

- A bug in which an unnatural energy distribution is tallied was fixed by setting `axis=eng` when EGS5 is used.
- The muon-nuclear interaction model used to implement muon capture reactions was revised.
- The default setting of the nuclear reaction model when light ions are targets was fixed. When such a reaction occurs, PHITS calculates it by assuming that the light ions are projectiles on the basis of inverse kinematics. For example, in the default setting INCL is used even for heavy ion induced reactions when the deuteron is as the target nucleus.

From version 2.76, the following functions were implemented. (2015/03/23)

- A muon-nuclear interaction model based on virtual photon production theory was implemented. Characteristic X-ray production from muonic atoms as well as associated muon capture reactions can be also modeled in this new version.
- Adjustment parameters for determining the magnitude of angular straggling for `nspred=2` were introduced.
- Bugs arising from the use of Intel Fortran 2015 were fixed.

From ver. 2.75, a bug in which the sumtally function does not work when using an input file that includes some sections of tally was fixed; additionally, a bug occurring when setting `e-mode=2` was corrected. (2015/02/09)

From version 2.74, the following functions were implemented. (2015/01/30)

- The version of DCHAIN included in the PHITS package was changed from DCHAIN2001 (dchain264.exe) to DCHAIN2014 (dchain274.exe). DCHAIN2014 was improved relative to DCHAIN2001 as follows:

(1) The input format was changed.

(2) The number of energy groups of neutron activation cross section libraries was increased from 175 to 1968.

(3) A new function was implemented to output the [source] section of PHITS from the activities calculated by DCHAIN.

(4) In the input format of ANGEL, a new function was implemented to output the time dependence of radioactivities in each region.

- Thread parallelization is now available even when using EGS5, i.e., `negs=1`. Some bugs related to EGS5 were also fixed. These improvements were performed by Mr. Masaaki Adachi of the Research Organization for Information Science & Technology (RIST) and was supported by the Center for Computational Science & e-Systems, Japan Atomic Energy Agency (JAEA).

- A new function to combine two (or more) tally results ("sumtally"), was implemented (from ver. 2.88, this function became applicable to all tallies): see Sec. 6.8 for more detail. This function was developed by Mr. Takamitsu Miura of RIST and was supported by the Center for Computational Science & e-Systems, JAEA.

- The Kurotama model was revised to enable calculation of cross sections over 5 GeV/u. See reference [9] shown below for more detail.

- The gamma de-excitation data contained in `trxcrd.dat` were incorporated into the source files of PHITS. Consequently, the parameter `file(14)` does not need to be specified in the PHITS input file even when setting `e-mode`≥1 or `|igamma|`≥1.

- Some bugs related to JAM and JAMQMD, etc., were revised.

From ver. 2.73, a bug producing abnormal nuclei such as di-neutrons in the calculation of nuclear reaction models was fixed. An installed executable file of the OpenMP version of this fix is available for Windows platforms (64-bit only); although PHITS can be executed via single processing on both 32-bit and 64-bit systems, it cannot be executed on 32-bit OpenMP. (2014/11/05)

From ver. 2.72, a bug occurring in the setting `igamma=2` was fixed and an error in which the GEM model produces di-neutrons was repaired. Furthermore, an error in the definition of angular distribution using degree was corrected for `a-type` in the `[source]` section. In the previous version, an incorrect interpolation caused the use of a biased distribution in setting the `a-type` sub-section using degree. Finally, the definition of `na` and `nn` in `[source]` using `a-type` was changed to prevent these parameters from being set to negative values. (2014/10/21)

From ver. 2.71, a bug with respect to electron-positron annihilation under EGS5 was corrected. (2014/09/26)

From ver. 2.70, the following functions have been implemented. (2014/08/30)

- The transport algorithm incorporated in EGS5 (Electron Gamma Shower Version 5 [10]) for photon, electrons, and positrons has been implemented in the PHITS code. This algorithm can be activated instead of the original by setting `negs=1` and in the `[parameters]` section and specifying `file(20)`. Currently, its OpenMP version is not available yet, and there is a limit of 100 materials that can be used at maximum when `negs=1`. (From ver. 2.80, there is no such limitation.) See Sec. 5.2.20 for details. This improvement was supported by Dr. Hirayama and Dr. Namito of KEK.

- High-energy photo-nuclear reactions of up to 1 TeV can be treated by implementing non-resonant the photo-nuclear reaction mechanism in JAM.

- Muon-induced nuclear reactions of up to 1 TeV can be treated by considering the generation of virtual photons from muons. This model can be activated by setting `imuint=1` in the `[parameters]` section.

[9] L. Sihver *et al.*, Nucl. Instr. & Meth. B 334, 34-39 (2014).

[10] H. Hirayama *et al.*, SLAC-R-730 (2005) and KEK Report 2005-8 (2005).

- The event generator mode ver.2 was improved to precisely determine the charged particle spectra on the basis of cross sectional data such as (n, p) and (n, α) contained in the evaluated nuclear data library. This new event generator mode can be used by setting `e-mode=2` in the `[parameters]` section.
- JQMD was improved to take relativistic effects into account, and an algorithm for stabilizing the initial states of nuclei was implemented. The improved JQMD, called JQMD-2.0, can be activated by setting `irqmd=1` in the `[parameters]` section. This improvement was performed in collaboration with Dr. D. Mancusi at CEA/Saclay.
- Detector resolution can be considered in the event-by-event deposition energy calculation by using `[t-deposit]` with `output=deposit`.

From ver. 2.67, the following functions were implemented. (2014/05/22)

- A geometry check function was implemented. This function applies when specifying a tally for generating a two-dimensional view of the geometry; when double-defined or undefined regions are detected, they are painted onto the two-dimensional view. See Sec. 10 for details.
- An extension of the event generator mode (ver.2) was implemented, resulting in an improvement of the accuracy of event-by-event analysis for reactions induced by neutrons below 20 MeV. See Sec. 5.2.22 for details.
- A new parameter `infout` was added to control the selectivity of output information in `file(6) (D=phits.out)`.
- The current batch number appears on the console window in real time. Important error and warning messages such as “input data file for cross section directory does not exist” are also shown in the window.
- A cone shape can be used to specify source locations by setting `s-type=18, 19`.
- The `dumpall` and `dump` functions for `[t-cross]`, `[t-time]`, and `[t-product]` tallies can now be used in the restart calculation. To implement this revision, the rule for specifying file names was changed: results written in a configuration file (.cfg) in the former version of PHITS (before 2.66) are now output in a file specified by `file=***`, while dump data are output in another file named `***_dmp`.
- The total memory usage of PHITS (`mdas`) given in the `param.inc` file was increased to 120,000,000 (equivalent to 1GB) and the maximum number of lattices in a cell was increased to 25,000,000. Using this extension makes it possible to utilize detailed voxel phantoms such as ICRP phantom without recompiling the source code.

From ver. 2.66, the following functions were implemented. (2014/02/21)

- An algorithm for including discrete spectra calculated using the Distorted Wave Born Approximation (DWBA) was implemented. In several nuclear reactions induced by protons or deuterons, discrete peaks are added to neutron and proton spectra obtained by nuclear reaction models.
- Pion production processes in photo-nuclear reactions were included by implementing Δ and N^* resonances. Thus, PHITS2.66 can treat photo-nuclear reactions up to 1 GeV（from ver. 2.70, this model is available up to 1 TeV).
- Results in units of Gy can now be obtained in the `[t-heat]` tally. A bug in which “NaN” was detected in void regions was corrected.
- A bug occurring when setting `nm` negative in the `[source]` section using `e-type=2,3,5,6,7,12,15,16`, which specify the energy spectrum as a functional shape, was corrected. Additionally, a similar bug for `nn` in the cases where `a-type=5,6,15,16`, which specify the angular distribution by shape, was also fixed.

From ver. 2.65, doses in units of Gy can be obtained in the `[t-deposit]` tally. Furthermore, a bug in converting mass density to particle density in the `[material]` and `[cell]` sections was fixed. This bug caused maximum errors of 0.6% in calculated results when neutron-rich nuclei were used. (2014/01/30)

From ver. 2.64, bugs in the photo-nuclear reaction model and in EBITEM, as well as other minor bugs, were fixed. Furthermore, a bug in which “NaN” was detected in `[t-heat]` calculations because of negative values in the probability table (p-table) was corrected. The Ace libraries were reproduced by neglecting the p-tables for the following 130 nuclides:

```
As075 Ba130 Ba132 Ba134 Ba135 Ba136 Ba137 Ba140 Br079 Br081 Cd106 Cd108
Cd110 Cd111 Cd112 Cd113 Cd114 Cd116 Ce141 Ce142 Ce143 Ce144 Cf250 Fe059
Ga069 Ga071 Hf174 Hf176 Hf177 Hf178 Hf179 Hf180 Hf181 Hf182 I_127 I_129
I_130 I_131 I_135 In113 In115 Kr078 Kr080 Kr082 Kr083 Kr084 Kr085 La138
La139 La140 Mo092 Mo094 Mo095 Mo096 Mo097 Mo098 Mo099 Mo100 Nb094 Nb095
Ni059 Pr141 Pr143 Rb085 Rb086 Rb087 Rh103 Rh105 Ru096 Ru098 Ru099 Ru100
Ru101 Ru102 Ru103 Ru104 Ru105 Ru106 Sb121 Sb123 Sb124 Sb125 Sb126 Se074
Se076 Se077 Se078 Se079 Se080 Se082 Sr084 Sr086 Sr087 Sr088 Sr089 Sr090
Tc099 Te120 Te122 Te123 Te124 Te125 Te126 Te127m Te128 Te129m Te130 Te132
Xe124 Xe126 Xe128 Xe129 Xe130 Xe131 Xe132 Xe133 Xe134 Xe135 Y_089 Y_090
Y_091 Yb168 Yb170 Yb171 Yb172 Yb173 Yb174 Yb176 Zr093 Zr095
```

(2013/11/19)

From ver. 2.60, the following functions were implemented. (2013/08/22)

- The algorithm for the de-excitation of nuclei after the evaporation process was improved by implementing the ENSDF-Based Isomeric Transition and isomEr production Model (EBITEM). Prompt gamma spectra including discrete peaks can now be precisely estimated, and isomer production rates can also be properly estimated.
- Quasi-deuteron disintegration, which is the dominant photo-nuclear mechanism from 25 to 140 MeV, was implemented in JQMD. Thus, PHITS2.60 can treat photo-nuclear reactions up to 140 MeV. (From ver. 2.70, this model is available up to 1 TeV.) The evaporation process after the giant resonances of ^{6}Li, ^{12}C, ^{14}N, and ^{16}O was improved by considering the isospins of excited nuclei, allowing alpha emission to be suppressed and neutron and proton emission to be enhanced via these giant resonances.
- Particle transport simulation in combined electro-magnetic fields became available: see section 5.11 for details.
- In the `[source]` section, several new energy mesh functions for directly defining differential energy spectra (in /MeV) and discrete energy spectra were implemented.
- Several algorithms were optimized to reduce the computational time, especially for the `xyz mesh` tally with `istdev=2`. Furthermore, use of memory for tally and ANGEL was improved. These improvements were performed by Mr. Daichi Obinata of Fujitsu Systems East Limited, and were supported by the Center for Computational Science & e-Systems, Japan Atomic Energy Agency (JAEA).
- Minor revision and bug fix.
  - The number of cells acceptable in `[t-dchain]` was increased.
  - The references of PHITS and INCL were changed.
  - 7-digit cell IDs became acceptable.
  - The maximum `dmax` for electrons and positrons was increased from 1 to 10 GeV.
  - Restart calculation became available even when PHITS does not stop properly.
  - Lattice cells became acceptable in `[t-dchain]`.
  - Termination of PHITS as a result of strange errors in JAM is now averted.
  - A new multiplier function `k=-120` was added to weight the density.
  - Several minor bug were fixed in SMM involving the user defined tally, range calculation, transform, lost electron particles, random number generation for MPI, and δ-ray production.
  - Nuclear data for some nuclei were revised by the following the revision of JENDL-4.0.
  - A bug in reading the proton data library was fixed.

From ver. 2.52, the following functions were implemented. (2012/12/27)

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

- A new tally [t-dchain] was implemented to generate DCHAIN input files to calculate the time dependence of activation during and after irradiation: please see Sec. 7.14 for more detail.
- Several macro bodies –Right Elliptical Cylinder (REC), Truncated Right-angle Cone (TRC), Ellipsoid (ELL), and Wedge (WED)– were implemented.

From ver. 2.50, the following functions were implemented. (2012/9/25)

- The procedure for calculating statistical uncertainties was revised. A function to restart PHITS calculation based on the tally results obtained by past PHITS simulations was implemented to increase the history number when the number is not sufficient: please see Sec. 5.2.2 for more detail. This improvement was performed by Mr. Daichi Obinata of Fujitsu Systems East Limited and was supported by the Center for Computational Science & e-Systems, Japan Atomic Energy Agency (JAEA).
- Shared memory parallel computing using OpenMP architecture became available in PHITS, although some restrictions remain (see Sec. 12.2). For this purpose, the source code of PHITS was drastically revised; as a result, old Fortran compilers such as f77 and g77 can no longer be used for compiling PHITS. See Sec. 11 for details. This work was supported by Next-Generation Integrated Simulation of Living Matter, Strategic Programs for R&D of RIKEN and by the RIKEN Special Postdoctoral Researchers (SPDR) Program. To implement this improvement, we used K computer and the RIKEN Integrated Cluster of Clusters (RICC).
- The cross-section data for photo-nuclear reactions 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 and, therefore, it has low accuracy when calculating higher energy photo-nuclear reactions above 20 MeV.
- The Statistical Multi-fragmentation Model (SMM) was implemented to handle the statistical decay of highly-excited residual nuclei. SMM improves accuracy in calculating the production cross sections of light and medium-heavy fragments in heavy ion collisions.
- Intra-Nuclear Cascade of Liège (INCL) was implemented as the default model for simulating nuclear reactions induced by neutrons, protons, pions, deuterons, tritons, ^{3}He, and ^{4}He particles at intermediate energies. This improvement was supported by Dr. Joseph Cugnon of the University of Liège and Dr. Davide Mancusi, Dr. Alain Boudard, Dr. Jean-Christophe David, and Dr. Sylvie Leray of CEA/Saclay under a collaboration between CEA/Saclay and JAEA.
- The KUROTAMA model, which gives nucleon-nucleus and nucleus-nucleus reaction cross sections, 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.
- The Intra-Nuclear Cascade with Emission of Light Fragment (INC-ELF) tool was implemented based on the work of the Uozumi research group under a collaboration between Kyushu University and JAEA.
- The user-defined [t-userdefined] tally was introduced to deduce user specific quantities from PHITS simulations. PHITS must be recompiled to use this tally; see Sec. 7.18 for details.
- The neutron kerma factors for several nuclei, such as ^{35}Cl, were revised. The photo- and electro-atomic data libraries were recreated based on JENDL-4.0 and the Livermore Evaluated Electron Data Library (EEDL), respectively.

From ver. 2.30, the radiation damage model for calculating Displacement Per Atom (DPA) in PHITS was improved through the application of screened Coulomb scattering. The [multiplier] section for use in the [t-track] section was also added. (2011/8/18)

From ver. 2.28, it is possible to use the dumpall and dump options for [t-cross], [t-time], and [t-product] tallies in 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 under parallel computing.

From ver. 2.26, a function to generate knocked-out electrons (so-called δ-rays) produced along the trajectories of charged particles was added. Setting the threshold energy for each region in the [delta ray] section enables the explicit transport of δ-rays above the threshold energy.

1.2 Development members

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

Tatsuhiko Sato, Yosuke Iwamoto, Shintaro Hashimoto, Tatsuhiko Ogawa, Takuya Furuta, Shinichiro Abe,
Takeshi Kai, Pi-En Tsai, Norihiro Matsuda, Yusuke Matsuya, and Hunter Ratliff
Japan Atomic Energy Agency (JAEA).

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

Nobuhiro Shigyo,
Kyushu University.

Lembit Sihver,
Vienna University of Technology, Austria.

The following members also contributed to the development of PHITS.

Hiroshi Takada, Shin-ichro Meigo, Makoto Teshigawara, Fujio Maekawa, Masahide Harada, Yujiro Ikeda,
Yukio Sakamoto, Hiroshi Nakashima, Tokio Fukahori, Keisuke Okumura, Tetsuya Kai, and Shusaku Noda,
Japan Atomic Energy Agency (JAEA).

Satoshi Chiba,
Tokyo Institute of Technology (TITech).

Takashi Nakamura,
Tohoku University.

Davide Mancusi,
Chalmers University, Sweden.

1.3 Reference of PHITS

Please refer the following document in context of using any version of PHITS.

- T. Sato, Y. Iwamoto, S. Hashimoto, T. Ogawa, T. Furuta, S. Abe, T. Kai, P.-E. Tsai, N. Matsuda, H. Iwase, H. Shigyo, L. Sihver, and K. Niita, Features of Particle and Heavy Ion Transport code System (PHITS) version 3.02, J. Nucl. Sci. Technol. 55, 684-690 (2018).

This is an open access article can be downloaded from `https://www.tandfonline.com/doi/full/10.1080/00223131.2017.1419890`

Other articles that describe the features of PHITS are:

- H. Iwase, K. Niita, T.Nakamura, Development of general purpose particle and heavy ion transport Monte Carlo code. J Nucl Sci Technol. 39, 1142-1151 (2002).

- K. Niita, T. Sato, H. Iwase, H. Nose, H. Nakashima and L. Sihver, Particle and Heavy Ion Transport Code System; PHITS, Radiat. Meas. 41, 1080-1090 (2006).

- L. Sihver, D. Mancusi, T. Sato, K. Niita, H. Iwase, Y. Iwamoto, N. Matsuda, H. Nakashima, Y. Sakamoto, Recent developments and benchmarking of the PHITS code, Adv. Space Res. 40, 1320-1331 (2007).

- L. Sihver, T. Sato, K. Gustafsson, D. Mancusi, H. Iwase, K. Niita, H. Nakashima, Y. Sakamoto, Y. Iwamoto and N. Matsuda, An update about recent developments of the PHITS code, Adv. Space Res. 45, 892-899 (2010).

- K. Niita, N. Matsuda, Y. Iwamoto, H. Iwase, T. Sato, H. Nakashima, Y. Sakamoto and L. Sihver, PHITS: Particle and Heavy Ion Transport code System, Version 2.23, JAEA-Data/Code 2010-022 (2010).

- K. Niita, H. Iwase, T. Sato, Y. Iwamoto, N. Matsuda, Y. Sakamoto, H. Nakashima, D. Mancusi and L. Sihver, Recent developments of the PHITS code, Prog. Nucl. Sci. Technol. 1, 1-6 (2011).

- T. Sato, K. Niita, N. Matsuda, S. Hashimoto, Y. Iwamoto, S. Noda, T. Ogawa, H. Iwase, H. Nakashima, T. Fukahori, K. Okumura, T. Kai, S. Chiba, T. Furuta and L. Sihver, Particle and Heavy Ion Transport Code System PHITS, Version 2.52, J. Nucl. Sci. Technol. 50:9, 913-923 (2013).

2 Installation and execution of PHITS

PHITS is a fortran-based code; it can be executed on Windows (7 or later), Mac (OS X v10.6 or later), Linux, and Unix operating systems. The installers are prepared for each environment. The execution environment can be easily setup through the use of installer. Note that an administrator account is required to use PHITS on Windows and Mac operating systems.

The recommended system requirements for PHITS are 2 GB of RAM and 6 GB (with at least 4 GB required) of available space on the hard disk.

The installation and execution methods for Windows, Mac, and Linux OS are respectively described in Sections 2.1, 2.2, 2.3. In addition, Section 3 describes how to resume PHITS calculations, and how to terminate the PHITS code during calculations.

2.1 Installation and execution on Windows OS

2.1.1 Installation on Windows OS

(1) If a previous version of PHITS has been installed, rename the existing folder to "phits-old" or something similar.

(2) Double click "setup-eng.vbs."

(3) Specify the installation folder ("c:\" is recommended).

(4) Enter the password.

(5) Installation is finished.

(6) To check whether PHITS has been installed properly, right click "\phits\lecture\basic\lec01\lec01.inp" and select "send to" → "PHITS."

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

(Note #1) The names of the installation files and folders and must contain no space characters; otherwise installation and execution will fail.

(Note #2) If you enter an incorrect password, please delete the "phits" folder first, and then run the installer again.

The installer "setup-eng.vbs" performs the following processes:

(1) It extracts "phits.zip" into the specified installation folder.

(2) It adds "\phits\bin\" in the environment variable "PATH."

(3) It creates shortcuts for three batch files—"phits.bat" and "angel.bat" in the "\phits\bin\" folder and "dchain.bat" in the \dchain-sp\bin\" folder— in the "sendto" folder.

(4) It revises the first line of the nuclear data list file "xsdir.jnd" in the "\phits \data\" folder as `datapath`= 'the installation folder'+ '\phits\XS\'.

2.1.2 Execution on Windows OS

The PHITS code can be executed either via File Explorer (previously known as Windows Explorer) using "sendto → PHITS" command, or via Command Prompt.

Execute using “send to → PHITS”

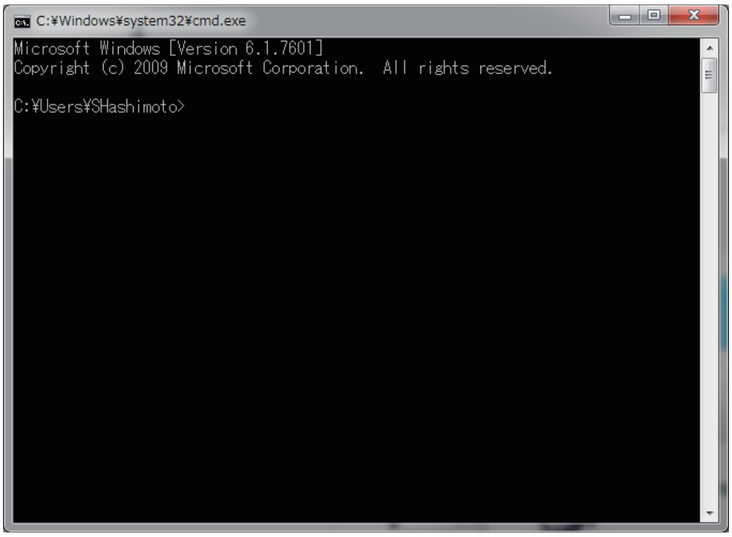
Right click the PHITS input file in a File Explorer, select “send to → PHITS” . As the shortcut for “phits.bat” in the “\phits\bin\” folder has been created during installation, the PHITS code will be initiated and start simulations.

The shared-memory parallel computing using OpenMP can be executed by adding `$OMP=`N (N is the number of CPU cores to be used) before the first section in the PHITS input file. If $N = 0$, all cores in the computer are used. If $N = 1$, the parallel computing is not used. From version 2.73, the executable file of the OpenMP version coming along with the PHITS package is available only for 64-bit Windows systems.

The MPI version of PHITS can be also executed by added `$MPI=`M in the same manner as OpenMP, but the installation of a MPI protocol (MPICH2) is required. Please see Sec. 12.1.1 in more detail.

Execute via Command Prompt

PHITS can be executed by command lines in Command Prompt (also known as cmd.exe or cmd), the command-line interpreter on Windows operating systems as shown in Figure 2.1.



```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:¥Users¥SHashimoto>
```

Figure 2.1: Example of how to execute PHITS via Command Prompt

To execute PHITS, use the “cd” command to go to the directory of where the input is, and then type the following commands in the terminal:

```
phits.bat your_input.inp
```

where “your_input.inp” is your input file name (e.g., “lec01.inp”). As command histories can be retrieved by pressing the ↑ key in the terminal, it is convenient to use the same input file name when you want to repeatedly execute PHITS.

2.1.3 Recommended software

There is no software that needs to be pre-installed before using PHITS. Nevertheless, it is recommended to install a text editor that can display line numbers, as the line number is specified when an error occurs in the PHITS input file. Furthermore, installation of Ghostscript and GSview (ver. 5 is preferable), which are written in the EPS format, is necessary to display image files created by PHITS.

Free text editors for Windows:

- Crimson Editor (`http://www.crimsoneditor.com/`).
- Notepad++ (`https://notepad-plus-plus.org/`).

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

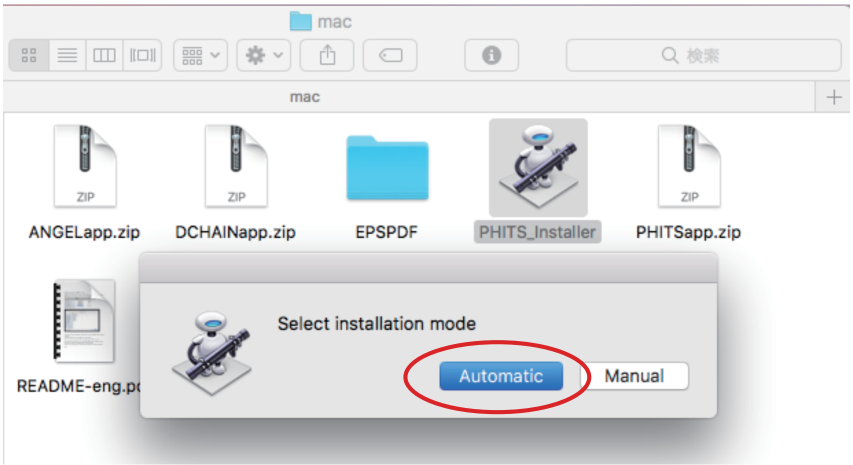
- Ghostscript (`http://www.ghostscript.com/`)
- GSview (`http://pages.cs.wisc.edu/~ghost/gsview/index.htm`).

2.2 Installation and Execution on Mac OS

2.2.1 Installation on Mac OS

(1) Double-click "phits_installer" included in the "/mac/" folder of the PHITS package.

(2) Select "Automatic" for "select installation mode" (See Fig. 2.2). "Manual" should be selected only when the "Automatic" mode fails. In that case, please refer to the "README-eng.pdf" file in the "/mac/" folder.



Select installation mode

Figure 2.2: Selection of installation mode.

(3) Specify the folder to install PHITS in. It is recommended to select the folder with user's account name (eg, iwamoto). Then click the "choose" button (See Fig. 2.3).

(4) A "/phits/" folder will be created under the account name. All the code content, including the executable files, source code, and documentation (manuals, lecture notes, input examples) will be copied to the specified folder; the process may take several minutes.

(Note #1) If a wrong password is typed during the installation, the message of "installation is completed" will immediately pop out, and the PHITS icon will appear on the Dock. However, no file is copied to the hard disk, and PHITS cannot be executed. In that case, please go back to the procedure (1) and type the correct password.

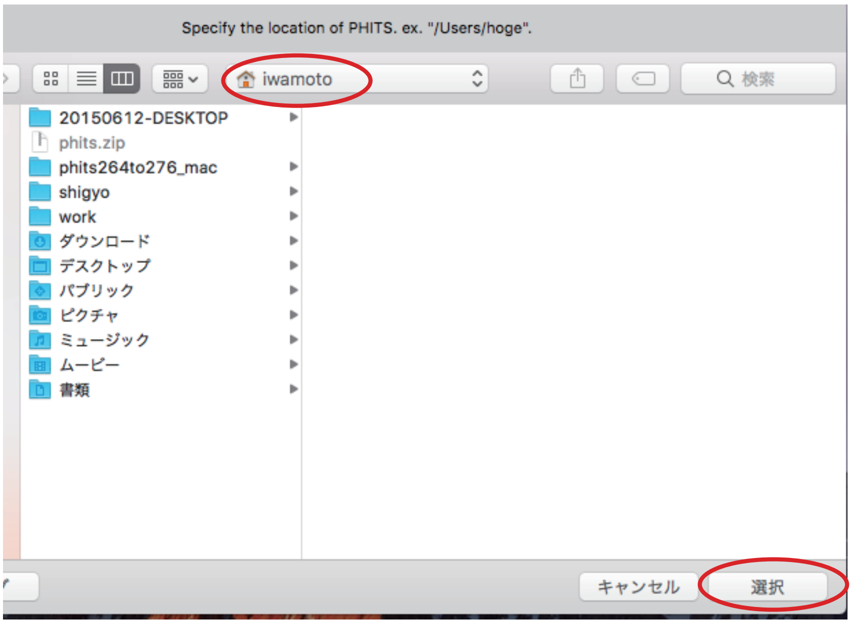
(Note #2) If the "/phits/" folder is moved to another folder after installation, PHITS will not work.

(Note #3) If a "/phits/" folder already exists in the specified folder, the old "/phits/" folder will be automatically renamed from "/phits/" to "/phits[date.time]/" of that moment.

(Note #4) If the names of the files and folders to be installed contain a blank character, the PHITS installation will fail.

2.2.2 Execution on Mac OS

PHITS can be executed by dragging and dropping or through the use of Terminal.



Specify the location of PHITS. ex. "/Users/hoge".

Figure 2.3: Screenshot of what folder is recommended for PHITS installation.

Execute by drag and drop

Drag the icon of the PHITS input file and drop it on the blue PHITS icon on the Dock (See Fig. 2.4). For test calculation, click the Finder icon on the Dock and open the folder of "/phits/lecture/basic/lec01/." Then, drag "lec01.inp" and drop it on the PHITS icon. A new terminal window will appear, and the calculation condition will be outputted on it. The calculated results will be shown in the same folder as the input file. As command histories can be retrieved by pressing the ↑ key in the terminal, it is convenient to use the same input file name when you want to repeatedly execute PHITS.

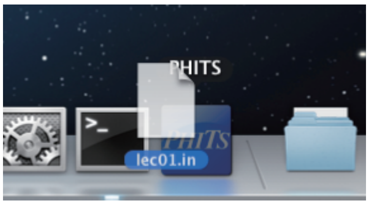


Figure 2.4: Execution by drag and drop.

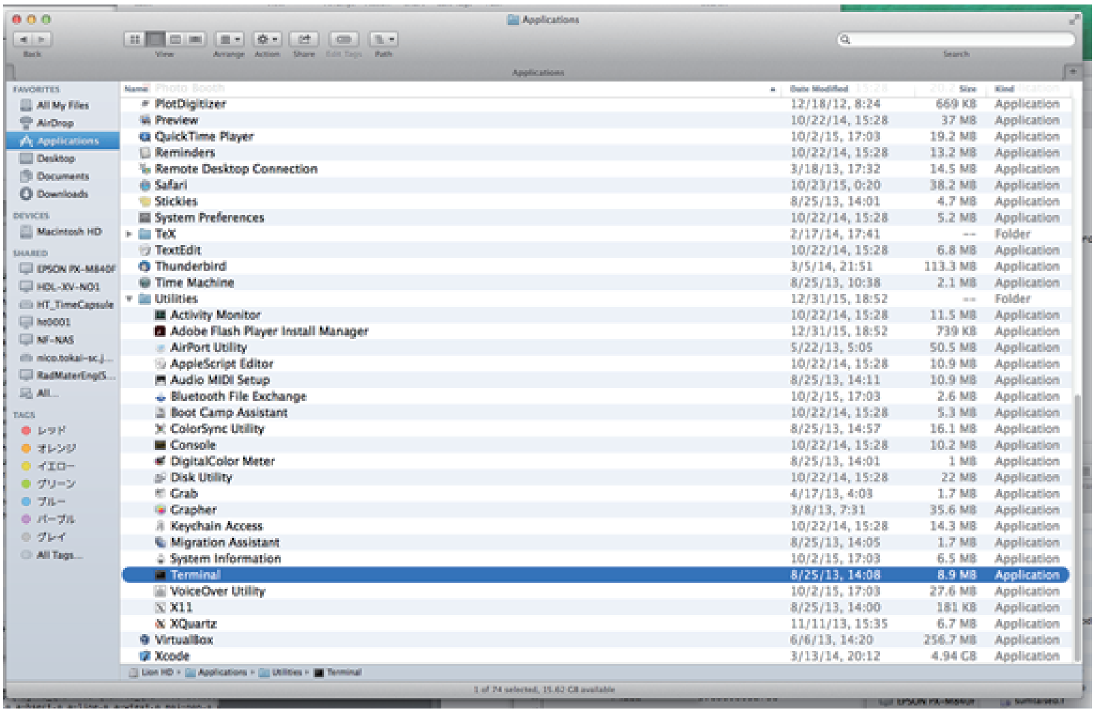
The shared-memory parallel computing using OpenMP can be executed by adding `$OMP=`N (N is the number of CPU cores to be used) before the first section in a PHITS input file. If $N = 0$, all cores in the computer are used. If $N = 1$, the parallel computing is not used.

To execute ANGEL or DCHAIN, drag the icons of the output file generated by the PHITS tally and drop it onto the red ANGEL or the green DCHAIN icon on the Dock. Note that the PHITS icon can identify whether the input file is in PHITS, ANGEL, or DCHAIN format, so dragging an ANGEL or DCHAIN input to the PHITS icon would also work.

Execute via Terminal

After installing PHITS by using "PHITS_Installer,"PHITS can be executed via terminal commands. To launch terminal, select the following items:

Finder → Applications → Utilities → Terminal



Name	Date Modified	Size	Kind
PlotDigitizer	12/18/12, 8:24	669 KB	Application
Preview	10/22/14, 15:28	37 MB	Application
QuickTime Player	10/2/15, 17:03	19.2 MB	Application
Reminders	10/22/14, 15:28	13.2 MB	Application
Remote Desktop Connection	3/18/13, 17:32	14.5 MB	Application
Safari	10/23/15, 0:20	38.2 MB	Application
Stickies	8/25/13, 14:01	4.7 MB	Application
System Preferences	10/22/14, 15:28	5.2 MB	Application
TeX	2/17/14, 17:41	--	Folder
TextEdit	10/22/14, 15:28	6.8 MB	Application
Thunderbird	3/5/14, 21:51	113.3 MB	Application
Time Machine	8/25/13, 10:38	2.1 MB	Application
Utilities	12/31/15, 18:52	--	Folder
Activity Monitor	10/22/14, 15:28	11.5 MB	Application
Adobe Flash Player Install Manager	12/31/15, 18:52	739 KB	Application
AirPort Utility	5/22/13, 5:05	50.5 MB	Application
AppleScript Editor	10/22/14, 15:28	10.9 MB	Application
Audio MIDI Setup	8/25/13, 14:11	10.9 MB	Application
Bluetooth File Exchange	10/2/15, 17:03	2.6 MB	Application
Boot Camp Assistant	10/22/14, 15:28	5.3 MB	Application
ColorSync Utility	8/25/13, 14:57	16.1 MB	Application
Console	10/22/14, 15:28	10.2 MB	Application
DigitalColor Meter	8/25/13, 14:01	1 MB	Application
Disk Utility	10/22/14, 15:28	22 MB	Application
Grab	4/17/13, 4:03	1.7 MB	Application
Grapher	3/8/13, 7:31	35.6 MB	Application
Keychain Access	10/22/14, 15:28	14.3 MB	Application
Migration Assistant	8/25/13, 14:05	1.7 MB	Application
System Information	10/2/15, 17:03	6.5 MB	Application
Terminal	8/25/13, 14:08	8.9 MB	Application
VoiceOver Utility	10/2/15, 17:03	27.6 MB	Application
X11	8/25/13, 14:00	181 KB	Application
XQuartz	11/11/13, 15:35	6.7 MB	Application
VirtualBox	6/6/13, 14:20	256.7 MB	Application
Xcode	3/13/14, 20:12	4.94 GB	Application

Lion HD ▸ Applications ▸ Utilities ▸ Terminal

1 of 74 selected, 15.62 GB available

Figure 2.5: How to launch Terminal: Finder → Applications → Utilities → Terminal.

To execute PHITS, use the “cd” command to go to the directory of where the input is, and then type the following commands in the terminal:

```
phits.sh your_input.inp
```

where “your_input.inp” is your input file name (e.g., “lec01.inp”). As command histories can be retrieved by pressing the ↑ key in the terminal, it is convenient to use the same input file name when you want to repeatedly execute PHITS.

ANGEL and DCHAIN can also be executed via the terminal. To execute ANGEL, type the following command in the Terminal:

```
angel_mac.sh angel.inp
```

where “angel.inp” is the input file name (tally output of PHITS. e.g., “track_xz.out”).

To execute DCHAIN, type the following command in the terminal:

```
dchain.sh dchain.out
```

where “dchain.out” is the name of the DCHAIN input file (the file name is designated in the `[t-dchain]` section of the PHITS input).

If the PHITS installation fails

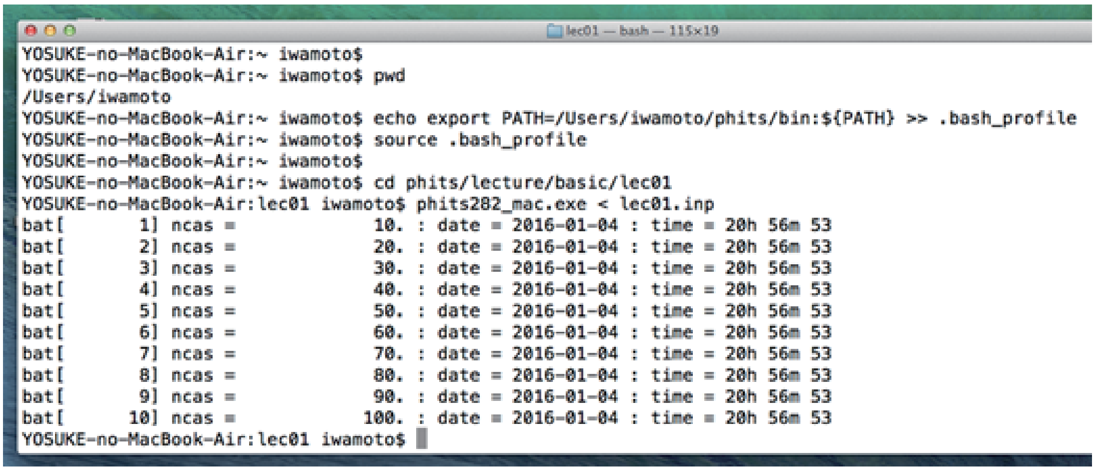
In such case, PHITS can still be executed via Terminal after some proper settings, even though it is not installed. Before executing PHITS via the Terminal for the first time, it is necessary to set PATH to the folder with the PHITS execution file. To do this, type the following commands in the Terminal:

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

where “/PATH-TO-PHITS” should be changed to the name of the folder of which PHITS is copied to (e.g., /Users/iwamoto/). If you do not know where the folder is, type in the following command in the Terminal:

```
find $HOME -name phitsXXX_mac.exe
```

where XXX is the version of PHITS. (e.g., 309. XXX is in the name of the PHITS package such as “PHITSXXX”). Note that there is no need to set PATH repeatly after executing PHITS the first time. `PHITSver=252` at the 8th line of the file “phits.sh” in the directory “/phits/bin/” should be changed to the version of PHITS that you would like to use. Hereafter, PHITS can be executed via terminal as shown the previous section of *Execute via Terminal*. Figure 2.6 shows an example of the initial setting for PHITS used in Terminal without installation, followed by a PHITS execution of a test input “lec01.inp.”



```
YOSUKE-no-MacBook-Air:~ iwamoto$
YOSUKE-no-MacBook-Air:~ iwamoto$ pwd
/Users/iwamoto
YOSUKE-no-MacBook-Air:~ iwamoto$ echo export PATH=/Users/iwamoto/phits/bin:${PATH} >> .bash_profile
YOSUKE-no-MacBook-Air:~ iwamoto$ source .bash_profile
YOSUKE-no-MacBook-Air:~ iwamoto$
YOSUKE-no-MacBook-Air:~ iwamoto$ cd phits/lecture/basic/lec01
YOSUKE-no-MacBook-Air:lec01 iwamoto$ phits282_mac.exe < lec01.inp
bat[       1] ncas =             10. : date = 2016-01-04 : time = 20h 56m 53
bat[       2] ncas =             20. : date = 2016-01-04 : time = 20h 56m 53
bat[       3] ncas =             30. : date = 2016-01-04 : time = 20h 56m 53
bat[       4] ncas =             40. : date = 2016-01-04 : time = 20h 56m 53
bat[       5] ncas =             50. : date = 2016-01-04 : time = 20h 56m 53
bat[       6] ncas =             60. : date = 2016-01-04 : time = 20h 56m 53
bat[       7] ncas =             70. : date = 2016-01-04 : time = 20h 56m 53
bat[       8] ncas =             80. : date = 2016-01-04 : time = 20h 56m 53
bat[       9] ncas =             90. : date = 2016-01-04 : time = 20h 56m 53
bat[      10] ncas =            100. : date = 2016-01-04 : time = 20h 56m 53
YOSUKE-no-MacBook-Air:lec01 iwamoto$ 
```

Figure 2.6: Example of PHITS execution via Terminal with the initial setting.

To execute DCHAIN, it is necessary to set PATH to the folder with the DCHAIN execution file. The method for setting the PATH for DCHAIN is the same as that for PHITS:

```
echo export PATH=/PATH-TO-PHITS/phits/dchain-sp/bin:${PATH} >> ˜/.bash_profile
source ˜/.bash_profile
```

where “/PATH-TO-PHITS” should be changed to the directory of which PHITS is copied to (e.g., /Users/iwamoto/).

2.3 Installation and Execution on Linux OS

2.3.1 Installation on Linux OS

Copy “phits.zip” from the PHITS installation package to a directory where you would like to install. Copy also the installer for Linux “PHITS-Installer_lin.sh” from the “linux” folder in the PHITS installation package to the same folder. Open a terminal and move to the installing folder. Then type the following command to add executable property to the installer:

```
chmod +x PHITS-Installer_lin.sh
```

Then execute the installer by the following command:

```
./PHITS-Installer_lin.sh
```

Installation will proceed in unzipping the ZIP file, modifying the installation PATH in files, and setting the execution PATH. Input of passward will be required during the unzip. PATH setting will be done in “.profile” with the assumption of the use of Bash. If you are using other shells, please change the setting accordingly. Installation is successfully finished if you do not see any error message.

Type the following command to activate the PATH setting:

```
source ˜/.profile
```

This command is not necessary anymore from the next login, since “.profile” will be automatically read when logging in.

2.3.2 Execution on Linux OS

In Linux environment, PHITS is executed via terminal. Open a terminal and move to the directory which contains the input file by the “cd” command. Then PHITS can be executed in the following fashion.

```
phits.sh your_input.inp
```

Here “your_input.inp” is the PHITS input file name (e.g., lec01.inp). As command histories can be retrieved by pressing the ↑ key in the terminal, it is convenient to use the same input file name when you want to repeatedly execute PHITS.

The shared-memory parallel computing using OpenMP can be executed by adding `$OMP=`N (N is the number of CPU cores to be used) before the first section in a PHITS input file. If $N = 0$, all cores in the computer are used. If $N = 1$, the parallel computing is not used.

The MPI version of PHITS can be also executed by added `$MPI=`M in the same manner as OpenMP, but it is required to pre-install OpenMPI. Please see the OpenMPI website (`https://www.open-mpi.org/`) for more detail.

The execution of ANGEL and DCHAN-SP can be done in a similar fashion. ANGEL can be executed by

```
angel.sh angel.inp
```

where “angel.inp” is the input file name (tally output of PHITS. e.g., “track_xz.out”).

To execute DCHAIN, type the following command in the terminal:

```
dchain.sh dchain.out
```

where “dchain.out” is the name of the DCHAIN input file (the file name is designated in the `[t-dchain]` section of the PHITS input).

Note that if errors, such as no permission, occur when executing PHITS, it might be because the execution permission has not been added to the executable file. For such a situation, go to “phits/bin/” or where the executable file is located in the terminal, and type the following command for authorization:

```
chmod +x XXX
```

Here “XXX” is the name of the executable file (e.g., phits309.exe).

Execute by specifying an executable file without using a shell

The PHITS code can be executed on Linux without using a shell by the following command:

```
phitsXXX.exe < your_input.inp
```

where “phitsXXX.exe” is the PHITS executable file and XXX is the version number, and “your_input.inp” is the input file for PHITS calculation.

(Note #1) If additional files are used with the `infl:` parameter when PHITS is executed with this method. In this case, the following text should be written in the first line of the main input file:

```
file = phits.inp
```

See Sec. 4.3 for further discussion of `infl:`.

(Note #2) To perform a PHITS calculation using distributed-memory parallel computing, the name of the input file must be specified in the first line of the “phits.in” file, which locates in the same folder containing the executable file, by the following manner:

```
file = input.inp
```

The name of “phits.in” is fixed and cannot be changed arbitrarily. See the example in Sec. 12.1.3 for more detail.

3 Restart and terminate PHITS calculations

3.1 Restart PHITS calculations

If tally results of PHITS calculations are not statistically significant enough, PHITS is able to read the previous tally results and resume the calculations (restart calculation mode). In the `[parameters]` section, restart calculation mode can be activated by setting `istdev < 0`, and the statistics of the restart calculation will be counted together with the previous tally results. It is allowed to add new tally sections for the input file with the restart calculation, even if there is no past tally result. Please refer to Table 5.2 for the setting of `istdev`.

The procedure of restart calculation is described as follows:

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

(2) If there is no `resfile` for a tally, it is regarded to be new. If no `resfile` can be found for any tally, a new calculation is begun with `istdev=|istdev|`.

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

(4) The tally parameters given in the current and past PHITS input files are then checked for consistency. If they are not consistent with each other, PHITS stops the calculation and outputs an error message. It should be noted that not all tally parameters are checked for consistency in this process.

(5) (In batch standard deviation mode only) The consistency of `istdev` and `maxcas` among the `resfiles` is checked. If they are consistent, the restart calculation is performed using these values. If an inconsistency is found, the calculation is stopped.

(6) The initial random seed is changed to the value of `rijklst` obtained from the first `resfiles`. If `rijklst` differs by `resfile`, a warning message is output.

(7) Upon finishing the restart calculation, the tally results are output to the file specified by `file=`. If `resfile` is not specified, the past tally results are overwritten.

Important notice:

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

II. The `maxcas` written in the input file is not used in the restart calculation in batch standard deviation mode.

III. Except for those given in the tally sections, the consistencies of input parameters are not checked by the PHITS program. If they are inconsistent with the previous setting, the tally results might be incorrect even if no error messages show up.

3.2 Terminate the PHITS code

Once PHITS is executed, it creates a "batch.out" [11] file following every batch that contains elapse information. In parallel calculation, batch.out also contains each processor element (PE) status. The occurrence of a PE abort can be determined by examining batch.out.

The first line of batch.out is written as follows:

```
10 <--- number of remaining batches
```

Here, the value on the left is the number of remaining batches at the moment. The remaining batches can be reduced by editing the value and saving batch.out. For example, when the value is changed to "3" the PHITS execution will be terminated after the calculation of three batches is finished. Changing the value to "0" terminates the code

[11] Until ver. 2.85, the name of this file had been named "batch.now."

execution right after the calculation of that batch is completed. This function can be useful in the way explained below.

The parameter itall can be used together with “batch.out”. If itall=0(default) or itall=1 is set in the [parameters] section, PHITS will update the calculation results (tally output) at the end of each batch. With itall=1, the image (*.eps) files will also be updated (See Sec. 6.7.17). In parallel computing mode, the results are updated every batch × (PE −1). Using these functions enables terminating PHITS calculations at any time with the latest results updated.

From ver. 2.86, the file name of batch.out can be modified by setting file(22) in the [parameters] section. Changing the name of each input file enables multiple executions of PHITS in a common directory. Note that this function is not available if [t-dchain] is set in the input files, because the same file name, “n.flux,” is used.

The value of rijk written in the batch.out file is the initial random number of the current batch. Thus, for example, in cases of unsuccessful termination of PHITS, it is possible to reproduce the calculation of the specified batch using the value of rijk.

3.3 Function to display transition of tally results and errors

When setting itall=3 in [parameters] section and “anatally” subsection in tally sections, transition of tally results and these statistical errors can be displayed as a function of the history number (or batch number). At the beginning of the calculation, there is a case in which the tally result or its error changes rapidly; we cannot decide whether the result is a reliable result or not when the error becomes very small. The purpose of this function is to confirm the reliability of the result by displaying the transition of each values as the history number increases.

Tally results and statistical errors at the end of each batch are output into a file ‘*_StD’ where ‘*’ indicates the file name specified in the tally. The values in the file are updated when the calculation of one batch finishes: the transition of the tally results until that time is shown.

This function is available only for [t-track] and [t-point] in the current version of PHITS.

3.3.1 Anatally subsection

When itall=3, mesh points to output tally results and these statistical errors can be specified by setting anatally subsection in each tally section.

Example 1: The case that mesh points are specified.

```
anatally start
ix = 89 91
iz = 61 81 101
iy = 1
ipart = 1
anatally end
```

“anatally” subsection should be defined between the lines of anatally start and anatally end written in the tally section. If anatally subsection has no content, meaning only these two lines are written, tally results at all mesh points are output.

ix,iz,iy are parameters to specify mesh points defined by x-type, z-type, y-type subsections, respectively. In this example, the 89th and 91st points of the x-mesh are specified. The specified points are given by the combination of mesh points of each parameter: in the above case, the six points, the product of two (89, 91) and three points (61, 81, 101), are specified. ipart=1 means the first particle set by part=. For example, only tally results for proton are output in the case that part = proton neutoron is set.

There are the following ten kinds of parameters to specify the mesh point: ireg: region (cell), ix, iy, iz, ir, ie, it, ia: x-, y-, z-, r-, e-, t-, a-type subsections, ipart: part parameter, and imul: multiplier parameter. When a parameter is not specified, results at all mesh points about its parameter are output. all can be used to output results at all points.

4 Input File

PHITS input consists of a number of sections that are listed in Tables 4.1 and 4.2. Each section begins with a [Section Name]. A maximum of four blanks can be placed between the line head and the declaration of [Section Name]; a configuration with (more than four blanks) [Section Name] will not be recognized as the beginning of a section, and the following code will be regarded as part of the previous section.

4.1 Sections

Table 4.1 and 4.2 shows the sections used in PHITS.

Table 4.1: Sections (1)

name	description
[title]	Title
[parameters]	Various types of parameters
[source]	Source definition
[material]	Material definition
[surface]	Surface definition
[cell]	Cell definition
[transform]	Definition of the coordinate transform
[temperature]	Cell temperature definition
[mat time change]	Definition of time-dependent materials
[magnetic field]	Magnetic field definition
[electro magnetic field]	Electro-magnetic field definition
[delta ray]	Definition of δ-ray production
[track structure]	Definition for using the track-structure simulation mode.
[super mirror]	Definition of super mirror for low-energy neutrons
[elastic option]	Definition of elastic scattering options for low-energy neutrons
[data max]	Definition of maximum energies (dmax) of each nucleus for using data libraries
[frag data]	Definition of user defined cross sections
[importance]	Definition of region's importances
[weight window]	Definition of region's weight windows
[ww bias]	Definition of values to bias in [weight window]
[forced collisions]	Forced collision definition
[volume]	Region volume definition
[multiplier]	Multiplier definition
[mat name color]	Material name and color definition for graphical plot
[reg name]	Region name definition for graphical plot
[counter]	Counter definition
[timer]	Timer and clock definition

Table 4.2: Sections (2)

name	description
[t-track]	Particle fluence in a given region.
[t-cross]	Particle fluence/current across a given surface.
[t-point]	Particle fluence at a given point.
[t-deposit]	Deposit energy in a given region.
[t-deposit2]	Deposit energies in two given regions.
[t-heat]	Heat generation in a given region. (Not recommended[12])
[t-yield]	Residual nuclei yields in a given region.
[t-product]	Reaction products in a given region.
[t-dpa]	Displacement Per Atom (DPA) in a given region.
[t-let]	LET distribution in a given region.
[t-sed]	Microdosimetric quantity distribution in a given region.
[t-time]	Time information of particle in a given region.
[t-interact] (former name [t-star])	Number of interactions occurred in a given region.
[t-dchain]	Residual nuclide yields (for the use of DCHAIN).
[t-wwg]	Output parameters for [weight window].
[t-wwbg]	Output parameters for [ww bias].
[t-volume]	Automatic calculation of the volume for a given region.
[t-userdefined]	Any quantities defined by users.
[t-gshow]	2D geometry visualization.
[t-rshow]	2D geometry visualization with physical quantities.
[t-3dshow]	3D geometry visualization.
[end]	End of input file.

Note that PHITS does not read any input information written below the [end] section.

4.2 Reading control

Uppercase, lowercase, blank(␣)

Discrimination between lowercase and uppercase characters is not performed in PHITS input except in file names. Blanks at the beginning and the end of a line are ignored except for the declaration of the [Section Name] as described above.

Tab

A tab is replaced by eight blanks.

Line connecting

The maximum number of characters that can be written in a line is 200. If '\' is added at the end of a line, the next line is considered to be a continued line. Multiple lines can be used to write input data using this method, but '\' is not necessary in the [surface] and [cell] sections, in which lines are automatically connected without the use of an additional symbol. Note that more than four blanks are required at the beginning of a continued line.

Line dividing

Several short lines can be displayed in one line, and divided by ';' as follows:

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

However, this function is not available when the format is defined in the mesh description.

Comment marks

The following comment marks can be used: #, %, !, $ (recommand). A comment out is effective starting from the comment mark to the end of the line. Note that in the [surface] and [cell] sections, only '$'

[12] Until ver. 3.04, the [t-heat] tally had been used to calculate deposit energy using the kerma approximation, because the [t-deposit] tally did not have the option.

can be used as a comment mark. Also, ‘c ’cannot be used as comment marks in the `[material]` section by the default setting[13]. To use ‘c’ as comment marks in the `[material]` section, it is necessary to set set `icommat=1` in `[parameters]` section[14].

Blank lines

Blank lines and lines beginning with a comment mark are skipped.

Section reading skip

Adding `off` after a section name (e.g., `[Section Name] off`) disables all the definitions in that given section (i.e., the section is being skipped). Note that `infl:` or `set:` written in the skipped section is also skipped.

Skipping lines within a section

To skip from any place in a section, put `qp:` at the line head. Lines from `qp:` to the end of the section are skipped.

Skip all

`q:` can be used as a terminator of an input file; this works the same way as using `[end]`.

4.3 Inserting files

Files can be placed in any location using:

```
infl:  { file.name } [ n1 − n2 ]
```

The name of the file to be inserted should be enclosed in curly brackets ‘{ },’ and the number of lines from n_1 to n_2 of the file should be enclosed in square brackets ‘[ ].’ If there is no ‘[ ],’ PHITS includes all lines of the specified file.

The following style can be used to specify line numbers: [n_1-] and [-n_2]. These expressions specify the range from the n_1th line to the end, and from the initial line to the n_2th line, respectively. Inserted files can be nested more than once. After reading the end of the including file, the reading process returns to its parent file.

Care should be taken in using the command-line interpreter (Command prompt) to execute PHITS. If `infl:` is used, the following text should be written in the first line of the input file:

```
file = phits.inp
```

Here, “phits.inp” is the input file name.

4.4 User-defined variables

User-defined variables can be set as follows:

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

The `set:` definition can be written anywhere. Note that there should be no space between `ci` and `[` in the format `ci[]`. User-defined variables can be used as numerical values in input file, and the variables can be re-defined at any time, with the values retained until they are re-defined. In the third case of the above example (`c3`), another variable `c1` is called in the definition; in this case, the value held by the variable `c1` at that time is used. Therefore,

[13] Prior to version 2.89, ‘c’ can be used as comment marks in the `[material]` section. If ‘c’ followed by a blank (␣) is placed before the 6th column, it is considered to be a comment line. Thus, if the natural isotope of carbon in the `[material]` section is defined as ‘C’, the definition line may be treated as a comment line and be skipped; in such case, carbon should be defined by `6000`.

[14] Note that the `[parameters]` section with `icommat=1` should be written above (before) the `[material]` section.

even if c1 is re-defined following the definition of c3, the value of c3 is not changed. Note that, by default, pi is set to the value of π.

The set: definition is ignored in sumtally subsection.

4.5 Using mathematical expressions

Mathematical expressions can be used in an input file following Fortran style guidelines. Available functions are shown in Table 4.3.

Table 4.3: Intrinsic Functions.

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

As an example of the use of standard mathematical expressions:

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

In the above example, because a single numerical value is expected after parameter=, blanks can be used in the expression. However, in some sections multiple numerical values cannot be accommodated; in such cases, an expression can be closed using '{ }', for example, { c1 * 2 / pi }.

4.6 Particle identification

The particle species that can be handled by PHITS are listed in Table 4.4. These particle species can be specified by their respective symbols or kf-codes. Particle species that are not assigned a symbol in Table 4.4 (itype=11) are specified only by their kf-codes. Table 4.5 lists the particles' decay-channels and their life-times.

Simulation of nuclei can be handled in PHITS with the use of physical models, such as JQMD. The isotope notation is in the format of the mass number followed by the chemical symbol of the element, e.g., 208Pb, 56Fe, etc.. When the mass number of an element is not specified (e.g., Pb and Fe), it refers to all naturally occurring isotopes of the respective element, and thus cannot be assigned as projectiles. Another way of describing nuclei is using the kf-code: `kf`= $Z * 1000000 + A$.

Table 4.4: List of 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[a]	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
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	$\Omega^-\ \overline{\Omega^-}$

[a] In the previous version of PHITS, `photon` was coded as `gamma`; in the new version, this is changed to `photon`.

Table 4.5: Decay channels and life times.

			branching ratio	life time (sec)
n	$\rightarrow$	$p + e^- + \overline{\nu}_\mathrm{e}$	100%	8.867×10^2
π^0	$\rightarrow$	$\gamma + \gamma$	100%	0
π^+	$\rightarrow$	$\mu^+ + \nu_\mu$	100%	2.6029×10^{-8}
π^-	$\rightarrow$	$\mu^- + \overline{\nu}_\mu$	100%	2.6029×10^{-8}
μ^+	$\rightarrow$	$e^+ + \nu_e + \overline{\nu}_\mu$	100%	2.19703×10^{-6}
μ^-	$\rightarrow$	$e^- + \overline{\nu}_e + \nu_\mu$	100%	2.19703×10^{-6}
K^0	$\rightarrow$	$\pi^+ + \pi^-$	68.61%	8.922×10^{-11}
	$\rightarrow$	$\pi^0 + \pi^0$	31.39%	
	$\rightarrow$	$\gamma + \gamma$	other	
K^+	$\rightarrow$	$\mu^+ + \nu_\mu$	63.51%	1.2371×10^{-8}
	$\rightarrow$	$\pi^+ + \pi^0$	other	
K^-	$\rightarrow$	$\mu^- + \overline{\nu}_\mu$	63.51%	1.2371×10^{-8}
	$\rightarrow$	$\pi^- + \pi^0$	other	
η	$\rightarrow$	$\gamma + \gamma$	38.9%	0
	$\rightarrow$	$\pi^0 + \pi^0 + \pi^0$	31.9%	
	$\rightarrow$	$\pi^+ + \pi^- + \pi^0$	23.7%	
	$\rightarrow$	$\pi^+ + \pi^- + \gamma$	other	
η'	$\rightarrow$	$\pi^+ + \pi^- + \eta$	44.1%	0
	$\rightarrow$	$\pi^0 + \pi^0 + \eta$	20.5%	
	$\rightarrow$	$\pi^+ + \pi^- + \gamma$	30.1%	
	$\rightarrow$	$\gamma + \gamma$	other	
Λ	$\rightarrow$	$p + \pi^-$	64.1%	2.631×10^{-10}
	$\rightarrow$	$n + \pi^0$	other	
Σ^+	$\rightarrow$	$p + \pi^0$	51.57%	7.99×10^{-11}
	$\rightarrow$	$n + \pi^+$	other	
Σ^0	$\rightarrow$	$\Lambda + \gamma$	100%	0
Σ^-	$\rightarrow$	$n + \pi^-$	100%	1.479×10^{-10}
Ξ^0	$\rightarrow$	$\Lambda + \pi^0$	100%	2.90×10^{-10}
Ξ^-	$\rightarrow$	$\Lambda + \pi^-$	100%	1.639×10^{-10}
Ω^-	$\rightarrow$	$\Lambda + K^-$	67.8%	8.22×10^{-11}
	$\rightarrow$	$\Xi^0 + \pi^-$	23.6%	
	$\rightarrow$	$\Xi^- + \pi^0$	other	

4.7 Array sizes

The array sizes described in the “param.inc” file may need to be checked and modified. The most important variable is `mdas`, which specifies the total size of arrays for geometry, tally output, nuclear data, and bank. To look up the current usage of the array, run an input echo (corresponding to `file(6)`=“phits.out”).

The bank size can be specified in the `[parameters]` section. When the bank runs out, an extra array of `mdas` is automatically used. The default param.inc is shown below.

File 1: param.inc

```
************************************************************************
*                                                                      *
*     'param.inc'                                                      *
*                                                                      *
************************************************************************

      parameter ( mdas   = 80000000 )
      parameter ( kvlmax =     3000 )
      parameter ( kvmmax =  1000000 )
      parameter ( itlmax =      200 )
      parameter ( inevt  =       70 )
      parameter ( isrc   =      500 )
      parameter ( latmax = 20000000 )
      parameter ( nbchmax=    10000 )

      common /mdasa/ das( mdas )
      common /mdasb/ mmmax

*----------------------------------------------------------------------*
*                                                                      *
*       mdas   : total memory * 8 = byte                               *
*       mmmax  : maximum number of total array                         *
*                                                                      *
*       kvlmax : maximum number of regions, cell and material          *
*       kvmmax : maximum number of id for regions, cel and material    *
*                                                                      *
*       itlmax : number of maximum tally entry                         *
*       inevt  : number of collision type for summary                  *
*       isrc   : number of multi-source                                *
*       latmax : maximum number of lattice in a cell + 1               *
*       nbchmax: maximum number of batch assigned to parallel MPI node *
*                                                                      *
*----------------------------------------------------------------------*
```

5 Sections format

5.1 [Title] section

In the `[title]` section, the titles of calculations can be defined. Any number of title lines is allowed. Blank lines are skipped in this section.

```
[ Title ]
This is a test calculation of PHITS.
Any number of title lines is allowed.
..........
```

5.2 [Parameters] section

The various parameters of PHITS can be defined in [Parameters] using the following format:

```
[ Parameters ]
para1 = number | file.name
para2 = number | file.name
..........
```

The order of parameters can be changed. As each parameter has a default value, undefined parameters use the default values.

Parameters and default values are listed in the tables shown below. (D=) indicates the default value.

5.2.1 Calculation mode

Table 5.1: Parameters (1)

Parameter	Value	Explanation
icntl	(D = 0)	Basic control option.
	= 0	Normal PHITS calculation.
	= 1	Nuclear reaction calculation, (under development).
	= 3	Output only input echo for checking memory usage, libraries, and file links.
	= 5	Calculation assuming all regions are void (no reaction, no ionization) for geometry check, and volume and area calculations.
	= 6	Source check. Source positions can be tallied by [t-product].
	= 7	Execute [t-gshow] for graphical output.
	= 8	Geometry output of xyz mesh tally with gshow option for graphical output.
	= 9	Execute [t-rshow] for graphical output.
	= 10	Geometry output of reg mesh tally with rshow option for graphical output.
	= 11	Execute [t-3dshow] for graphical output.
	= 12	Re-calculate using dumpall file. Dumpall file is specified by file(15).
	= 13	Use of function to sum up tally results (Sumtally function).
	= 14	Use of automatic calculation of region volume.
	= 15	Execute [t-wwbg] to obtain [ww bias] parameters.

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

By setting icntl=12, PHITS reads the previous event-by-event calculation result from a dumpall file created when the dumpall=1 option was activated. All of the transport events are reproduced by the dumpall file without being re-calculated again. Thus, maxcas and maxbch cannot be changed. This functionality is suitable for tallying different physics quantanties with the same simulation setup, especially when such a simulation setup is computational time consuming. However, it should be noted that calculation with dumpall=1 may create a huge dumpall file. Please refer to 5.2.21 for more details.

By setting icntl=13, two (or more) tally results can be summed: see Sec. 6.8 for further detail.

By setting icntl=14 with [t-volume], automatic calculations to obtain volumes of specified cells can be performed. See Sec. 7.17 in more detail.

By setting icntl=15 with [t-wwbg], automatic calculations to obtain parameters of [ww bias]. See Sec. 7.16 for further detail.

5.2.2 Number of history and Bank

Table 5.2: Parameters (2)

Parameter	Value	Explanation
irskip	(D = 0)	Random number control.
	irskip > 0	Skip the first irskip number of histories and the random number seeds for debugging. (The total number of history is reduced by irskip.)
	irskip < 0	Begin calculation with the No. irskip+1 random number seeds for executing parallel computing manually. (The total number of history is unchanged.)
rseed	(D = 0.0)	Initial random number option.
	rseed < 0	Get an 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 histories per batch. The upper limit is 2147483647.
maxbch	(D = 10)	Number of batches. The upper limit is 2147483647.
maxbnk	(D = 10000)	Number of particle temporary saved in a memory.
timeout	(D = -1.0)	Cut-off CPU time for PHITS calculation (in units of second).
istdev	(D = 0)	Control parameter for statistical uncertainty calculation type and restart mode.
	= -2	Restart calculation mode is activated; if there is no past tally result, a new calculation is started with istdev=2.
	= -1	Restart calculation mode is activated; if there is no past tally result, a new calculation is started with istdev=1.
	= 0	istdev is automatically set to 1 for shared-memory parallel computing, and 2 for the other cases.
	= 1	A new calculation is begun. Statistical uncertainties are estimated from the standard deviation of each batch result.
	= 2	A new calculation is begun. Statistical uncertainties are estimated from the standard deviation of each history result. This option cannot be selected in shared-memory parallel mode.
ireschk	(D = 0)	Control parameter for tally consistency check.
	= 0	Check consistency between new and old tally settings.
	= 1	No check. This option is useful when setting a very complex tally whose settings are not fully written in the tally output file.

In the distributed-memory parallel calculation, the number of batches should be an integer multiple of $N_{\mathrm{PE}} - 1$, where N_{PE} is the total number of Processing Element (PE). Otherwise, PHITS will automatically convert maxbch to an integer multiple of $N_{\mathrm{PE}} - 1$, and adjust maxcas such that the total number of histories becomes close to or the same as the given total number of histories. In this case, some comments will be output at the end of an input echo.

Time limit of a PHITS calculation can be activated by assigning timeout a positive value. If the CPU time reaches timeout in the middle of a batch, PHITS will finish the calculation of all histories in that batch, and then terminates the code. In the distributed-memory parallel computing, timeout is ruled by the sum of all CPU time. Note that the function of timeout can be used only when the computer's CPU time is correctly retrieved.

The procedure for calculating statistical uncertainties was revised from version 2.50: a function to restart PHITS calculation based on the tally results obtained by past PHITS simulations has been implemented to increase the total history number and then decrease the statistical uncertainties of the results. In this mode, the initial random seed is also read from the past tally file.

To calculate statistical uncertainties, two modes can be selected –"batch standard deviation mode" and "history standard deviation mode"– which calculate standard deviations of tally results by batch and by history, respectively.

In both modes, standard deviation σ is calculated as

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

where N is the total batch number (`istdev=1`) or the total history number (`istdev=2`),

x_i is the tally results,

w_i is the source weight of each sample,

$\overline{X}$ is the mean value of the tally results, and

$\overline{w}$ is the source weight of N samples.

The ratio of σ to $\overline{X}$ is written as relative error in the tally output file. In shared-memory parallel computing mode, only the batch standard deviation mode can be selected as it is impossible to calculate tallied quantities history-by-history using shared-memory parallel computing. The standard deviations calculated by the batch standard deviation mode vary by combination of `maxcas` and `maxbch`, even for the same total history number. In principle, a larger `maxbch` provides more reliable statistical uncertainties in batch standard deviation mode but may require a longer computational time. Setting the value of `maxbch` greater than 10 is recommended to obtain reliable results. By contrast, in history standard deviation mode, the standard deviation depends only on the total history number and is independent of the combination of `maxcas` and `maxbch`. Thus, it is recommended to select the history standard deviation mode (`istdev=2`) when memory-shared parallel computing is not in use. Note, however, that computational time occasionally becomes extremely long in history standard deviation mode, especially in cases in which there are tallies using many memories, e.g., `xyz` mesh tally with its very fine structure.

In tallies for calculating the standard deviation of deposition energies by individual history, such as `[t-deposit]` with `output=deposit` and `[t-deposit2]`, the standard errors instead of the standard deviations are output as relative errors. The relative standard errors can be estimated as $1/\sqrt{K}$ where K is the number of histories contributing to the tally result. This calculation procedure is independent of the `istdev` parameter.

Relative errors are generally output in the `r.err` column, which is the rightmost column of tally results. In the case of 2D-plots, such as tallies with `axis=xy` or `rz`, errors are output into another file, '*_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, a graph for the error in the 2D-plot can be obtained by ANGEL through a conversion process.

Note that as what PHITS calculates is the standard deviation or the standard error, it means the true value has a 68% of chance falling within the $\pm1\sigma$ confidence interval. Furthermore, PHITS does not estimate systematic uncertainties contributed by the nuclear reaction models.

When `istdev < 0`, the restart calculation mode is activated; however, if there is no past tally result, a new calculation is started with `istdev=|istdev|`; in other words, the batch and history standard deviation modes are selected for `istdev = -1` and `-2`, respectively. When restarting a calculation, the standard deviation mode is automatically determined from the past PHITS calculation. The tally parameters in the restart calculation must be consistent with the original tally parameters (i.e., parameters that already exist in the previous tally results must remain unchanged); however, new tally sections can be added in the restart calculation.

5.2.3 Cut-off energy and switching energy

Table 5.3: Parameters (3)

Parameter	Value	Explanation
emin(1)	(D = 1.0e-3)	Proton cut-off energy [MeV].
emin(2)	(D = 1.0e-3)	Neutron cut-off energy [MeV].
emin(i)	(D = 1.0e-3)	(i=3–10) Cut-off energy for pions, muons, and kaons [MeV]. (i=particle id; see Table 4.4)
emin(11)	(D = 2.0)	Cut-off energy for particle species with itype = 11 in Table 4.4 (MeV).
emin(i)	(D = 1.e+9)	Cut-off energy for electrons (i=12), positrons (i=13), and photons (i=14) [MeV].
emin(i)	(D = 1.0e-3)	Cut-off energy for deuterons (i=15), tritons (i=16), ^{3}He (i=17), ^{4}He (i=18), and nuclei (i=19)[MeV/u].
esmin esmax	(D = 0.001) (D = 300000)	Lower energy limit for range calculation of charged particles [MeV/u]. Upper energy limit for range calculation of charged particles [MeV/u]. Note that these parameters are not applicable for electrons and positrons.
cmin(i)	(D = emin(i) or 1.0) i= 15 – 19	Nuclear reaction cut-off energy for i-th particle [MeV]. Any nuclear reactions at energies below cmin(i) are not treated. For these nuclei, energy unit is [MeV/u].
dmax(i)	(D = emin(i))	Upper energy limit of library use for i-th particle [MeV]. Note that the user has to prepare the data library except for dmax(i=2,12,13,14), which are included in the PHITS package.
etsmin etsmax	(D = 1e-6) (D = 1e-2)	Lower energy limit of particles simulated by track structure mode [MeV]. Upper energy limit of particles simulated by track structure mode [MeV].

If values given by these parameters are referred to the upper boundaries of a range, then the given values are not included in the range. If values given by the parameters are referred to the lower boundaries of a range, then the given values are included in the range. For example, a proton right at the emin(1) energy is not cut-off as the cut-off range is $\geqslant 0$ and $<$ emin.

When the kinetic energy of a transport particle is less than emin, the energy cut-off is performed and then its transport calculation is stopped. In this case, the residual kinetic energy is deposited locally at that place, and the particle decays in accordance with decay modes shown in Table 4.5 except for neutrons. If the particle species is positron, annihilation occurs instead of decay.

PHITS uses libraries in the energy region emin < energy < dmax. If emin > dmax is set, no libraries are used. The upper energy limits for the use of data libraries are 20 MeV and 100 GeV, respectively for neutron and photon. When EGS5 is not used, the upper energy limits for electrons and positrons are 10 GeV. The default setting of emin(12-14) is automatically adjusted by the negs option; e.g., emin(14)=0.001 if negs=-1. The default settings of emin(2) and dmax(2) are automatically adjusted by the nucdata option; e.g., emin(2)=1.0e-11 if nucdata=1. See Table 5.5 for both settings in more details.

The range table of charged particles is set within esmin < energy < esmax. To use a significantly higher energy, the user should set esmax.

The minimum cut-off energy for charged particles, emin, cannot be set lower than esmin. In such cases, emin is automatically adjusted to esmin.

etsmin>1e-9 (1meV) can be set. But the setting below 1e-6 (1 eV) is not recommend because computational time becomes extremely long.

etsmax>1e-3 (1keV) must be set. The setting this parameter below 0.1 (100 keV) is recommend, otherwise the computational time becomes extremely long.

For the track structure mode, emin(12) and emin(13) should be set to 1.0e-3, and EGS5 should be activated (negs=1).

Below eqmdmin, the nuclear reactions of d, t,^{3}He, ^{4}He, and nuclei are not treated by JQMD. As the applicability of JQMD is restricted in the low energy region and the range of nuclei is very low in the normal material, it is not necessary to consider the low energy reactions of nuclei for the usual case. As a default, high energy heavy ion collisions are treated by JAMQMD above 3.0 GeV/u. This switching energy can be changed by changing e

Table 5.4: Parameters (4)

Parameter	Value	Explanation
`ejamnu`	`(D=20.)`	Switching energy of nucleon-nucleus reaction calculation from Bertini (or JQMD) to JAM model [MeV].
`ejampi`	`(D=20.)`	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=20.)`	Switching energy of nucleon-nucleus reaction calculation from Bertini to JQMD model [MeV/u].
`eqmdmin`	`(D=10.0)`	Minimum energy of JQMD calculation [MeV/u].
`ejamqmd`	`(D=3000.0)`	Switching energy from JQMD to JAMQMD [MeV/u].
`inclg`	`(D=1)`	Control parameter for use of INCL.
	`=0`	INCL is not used.
	`=1`	Use of INCL in a proton, neutron, pion, $d, t,^3$He, or ^{4}He 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`	INC-ELF is not used.
	`=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].
`irqmd`	`(D=0)`	Control parameter for use of JQMD or JQMD-2.0.
	`=0`	Use of JQMD in nuclear reactions.
	`=1`	Use of JQMD-2.0 in nuclear reactions.
`kerma`	`(D=0)`	Option for kerma mode.
	`=0`	This option is not used.
	`=1`	When neutron or proton nuclear data library of is used, PHITS produces no charged particles except for protons.

`jamqmd`. It is possible to calculate even nucleon-induced collisions in JAMQMD by changing `eqmdnu, ejamnu`, and `ejamqmd`.

INCL (Intra-Nuclear Cascade of Liège) is a nuclear reaction model for nucleon (proton and neutron), pion, and light-ion ($d, t,^3$He, or ^{4}He) induced reactions. From version 2.50, INCL is used by default for these reactions if the nuclear reaction model is not explicitly specified. Before using INCL results in a publication, please refer to a document [15] in the footnotes.

Intra-Nuclear Cascade with Emission of Light Fragment (INC-ELF) is a nuclear reaction model for nucleon-induced reactions. Before using results obtained by INC-ELF in a publication, please refer a document[16] in the footnotes.

JQMD and JQMD-2.0 are nuclear reaction models; in particular, they can be used to describe heavy-ion induced reactions. In PHITS Ver. 2.7 and later, JQMD-2.0 can be used as an alternative to the conventional JQMD. JQMD-2.0[17] describes reactions - particularly peripheral collisions - more reasonably than JQMD. Users should note that JQMD-2.0 may take more than twice the CPU time required by JQMD.

[15] A. Boudard, J. Cugnon, J.-C. David, S. Leray, and D. Mancusi, Phys. Rev C87, 014606 (2013).

[16] Y. Sawada, Y. Uozumi, S. Nogamine, T. Yamada, Y. Iwamoto, T. Sato, and K. Niita, Nucl. Instr. & Meth. B 291, 38-44 (2012).

[17] T. Ogawa, T. Sato, S. Hashimoto, D. Satoh, S. Tsuda, and K. Niita, Phys. Rev C92, 024614 (2015).

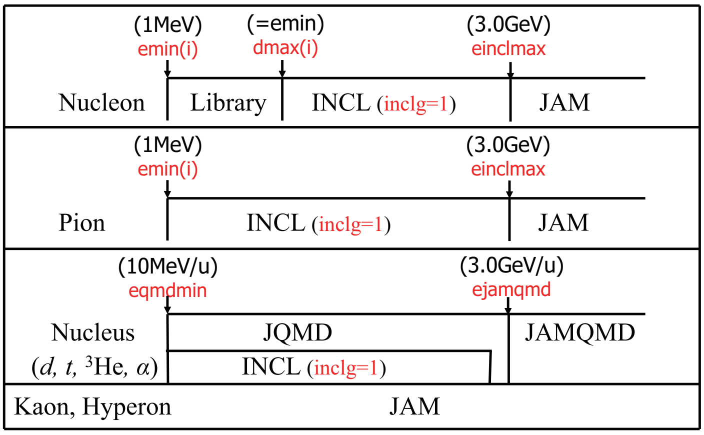


Figure 5.1: Map of Nuclear Reaction Models.

Table 5.5: Parameters (5)

Parameter	Value	Explanation
`negs`	(D=-1)	Option for electron, positron, and photon transport.
	=-1	Transport only photons based on the PHITS original algorithm. By selecting this option, `emin(14)` and `dmax(14)` are automatically set to 0.001 and 1000.0, respectively. If these parameters are directly specified in the input file, the specified parameters overwrite the setting.
	= 0	Ignore electron, positron, and photon transport.
	= 1	Transport electrons, positrons, and photons based on the EGS5 algorithm. By setting this option, `emin(12,13)` and `emin(14)` are automatically set to 0.1 and 0.001, respectively, while `dmax(12-14)` are set to 1000.0. If these parameters are directly specified in the input file, the specified parameters overwrite the setting. `file(1)` or `file(20)` must be specified in this option.
`nucdata`	(D=1)	Option for automatic setting of `emin(2)` and `dmax(2)` for the usage of nuclear data library.
	= 0	No adjustment (nuclear data are not used).
	= 1	Change the parameters suitable for JENDL-4.0, i.e. `emin(2)=1.0e-11` and `dmax(2)=20.0`. If these parameters are directly specified in the input file, the specified parameters overwrite the setting. `file(1)` or `file(7)` must be specified in this option.

5.2.4 Cut-off time, cut-off weight, and weight window

Table 5.6: Parameters (6)

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 4.4).

wc1(i)
wc2(i)
swtm(i)
	(D=-0.5)
	(D=wc1/2)
	(D=1.0)
		Minimum weight for i-th particle.
		Cut-off weight for i-th particle.
		Minimum source weight for i-th particle.

wupn	(D=5)	Maximum value of weight window is given by the product of wupn × (minimum value in [Weight Window] section). Here, wupn ≥ 2 should be set.
wsurvn	(0.6*wupn)	Survival weight value. Here, 1 < wsurvn < wupn should be set.
mxspln	(D=5)	Maximum number of split and maximum multiple number of survival. Here, mxspln > 1 should be set.
mwhere	(D=0)	Where the weight window takes place. -1: at nuclear reaction, 0: both, 1: at region crossing.
iwwbias	(D=0) = 0 = 1	Option for [ww bias]. Values defined in [ww bias] are not used. Values defined in [ww bias] are used. The [weight window] parameters multiplied by inverse of the defined biases in [ww bias] are used.
istdcut	(D=0) = 0 = 1	Option for controlling whether tally calculations should be stopped or not when their statistical uncertainties become less than stdcut. Note that the random numbers might be duplicated when you restart the PHITS simulation after terminating some tally calculations in the previous simulation. In the restart case, you have to write the tally lastly stopped in the previous simulation should be written at the top of the input file. Continue all tally calculations until the end of the PHITS simulation. Stop each tally calculation when its statistical uncertainty becomes less than stdcut.
istdbat	(D=0)	Inactivate stdcut until when batch number exceeds istdbat. This parameter is useful particularly for the OpenMP version because statistical uncertainties of tally results are fluctuated very much at the early stage of the PHITS simulation, due to the automatic selection of the batch variance (istdev = 1).

The cut-off time for each particle should be specified as tmax(i) (in units of [nsec]). After arrival at the cut-off time, the particle is killed; although this is not effective in the context of high energy particle transport, it is useful for low energy particle transport calculation.

A particle's weight is affected by the importance, forced collisions, implicit captures, and weight window functions. When the weight is lower than the user-defined weight cut off, a Russian roulette method is applied to determine whether or not the particle is killed. This function is not available for particles defined in the weight window.

Under the Russian roulette method, when the weight wgt is lower than the product, wc2× R, of wc2 and the ratio R of the importances at the source and current points (i.e., when wgt < wc2 × R), the particle survives with a probability wgt /(wc1×R), which is a function of the particle's own weight, WGT. If the particle survives, its weight is changed to wgt = wc1 × R. If wc1 and wc2 are negative, they are respectively set to |wc1| = swtm(i) and |wc2| = swtm(i).

Any particles or regions for which the user has not set importances are given default importances of 1.

In this case of iwwbias=1, the products of the multiplication are output in the input echo of [weight window], and [ww bias] with off is output. If an input file without [ww bias] is used, all values of [ww bias] in the input echo are set to 1.

5.2.5 Model options (1)

Table 5.7: Parameters (7)

Parameter	Value	Explanation
`ielas`	`(D=2)`	Elastic scattering option.
	`= 0`	Exclude elastic scattering.
	`= 1`	Include neutron elastic scattering.
	`= 2`	Include neutron and proton elastic scattering.
`ielms`	`(D=100)`	Number of angle groups for elastic scattering.
`inmed`	`(D=1)`	Nucleon-nucleon cross section options for Bertini model.
	`= 0`	Free (nmtclb25.dat).
	`= 1`	Cugnon old (nmtclb95.dat); the in-medium effect for nucleon-nucleon cross sections is included.
	`= 2`	Cugnon new (nmtclb30.dat).
`nevap`	`(D=3)`	Options for evaporation model.
	`= 0`	Without evaporation 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 the default value of 100 fm/c to 75 fm/c.
`ngem`	`(D=1)`	Options for the version of the GEM model.
	`= 0`	Use the default version (currently ver.1).
	`= 1`	Use Ver.1.
	`= 2`	Use Ver.2 (including nucleon-gamma emission competition、KTUY data[18] for shell correction).
`igamma`	`(D=2)`	γ decay options for residual nuclei[19] .
	`= 0`	Without γ decay.
	`= 1`	With γ decay.
	`= 2`	With γ decay based on the EBITEM model.
	`= 3`	With γ decay and isomer production based on the EBITEM model.
	`=-1,-2,-3`	Same as 1, 2, 3, except that Doppler effect owing to the motion of excited nuclei is disregarded. Do not use unless necessary, as it distorts physical reaction processes.
`isobar`	`(D=0)`	Options for the isobar model.
	`= 0`	Without isobar.
	`= 1`	With isobar.
`ipreeq`	`(D=0)`	Options for the pre-equilibrium model when `nevap=1`.
	`= 0`	Without the pre-equilibrium model.
	`= 1`	With the pre-equilibrium model.
`level`	`(D=3)`	Level density options when `nevap=1`.
	`= 1`	8/A.
	`= 2`	With Baba's parameters.
	`= 3`	With Ignatyuk's parameters.

With the Statistical Multi-fragmentation Model (SMM), PHITS improves the accuracy of calculating the production cross sections of light and medium-heavy fragments for collisions of heavy ions such as Pb or Hg or for incident energies over 100 MeV/u. It should be noted that the computational time becomes long when using this model: see the references [20] [21] for details.

Setting `igamma=3` enables obtaining of information on isomer production using `[t-yield]` with `axis=chart` or `dchain`: see reference [22] in the footnotes for details.

[18] H. Koura, T. Tachibana, M. Uno, and M. Yamada, RIKEN Accel. Prog. Rep. 36 (2003) 9: H. Koura, TOURS Symposium on Nuclear Physics V, AIP Proceedings 704 (2004) 60.

[19] Prior to ver. 2.73, `file(14)=trxcrd.dat` is required for `igamma=1-3`.

[20] T. Ogawa, T. Sato, S. Hashimoto, and K. Niita, Nuclear Instruments and Methods in Physics Research A 723 (2013) 36-46.

[21] J.P. Bondorf, A.S. Botvina, A.S. Iljinov, I.N. Mishustin, and K. Sneppen, Physics Reports 257 (1995) 133-221.

[22] T. Ogawa, S. Hashimoto, T. Sato, and K. Niita, Nuclear Instruments and Methods in Physics Research B 325 (2014) 35-42.

5.2.6 Model options (2)

Table 5.8: Parameters (8)

Parameter	Value	Explanation
ieleh	(D=0) = 0 = 1	Options for electron and positron transport. No slowing down and no reaction in the energy region above dmax(12). When the energy, e, is above dmax(12), the weight value, wgt, is changed to wgt= e/dmax(12), and then the transport calculation is performed with the energy of dmax(12).
ipnint	(D=0) = 0 = 1 = 2	Options for photo-nuclear reaction. [23] This reaction mechanism is not considered. This reaction mechanism is considered, except for nuclear resonance fluorescence. This reaction mechanism is considered, including nuclear resonance fluorescence. (See notes below this table for polarization of incident photons.)
pnimul	(D=1.0)	Multiplying factor to increase sampling probability of photo-nuclear reaction.
imucap	(D=1) = 0 = 1 = 2	Options for capture reactions of negative-muons considering characteristic X-ray production from muonic atoms and nuclear absorption. This reaction mechanism is not considered. This reaction mechanism is considered with default parameters used for calculation of characteristic X-ray production. This reaction mechanism is considered with user defined parameters written in file(28) used for calculation of characteristic X-ray production.
imuint	(D=1) = 0 = 1	Options for muon-induced nuclear reaction based on the virtual photon theory. This reaction mechanism is not considered. This reaction mechanism is considered.
imubrm	(D=1) = 0 = 1	Options for muon-induced bremsstrahlung. This reaction mechanism is not considered. This reaction mechanism is considered.
imuppd	(D=1) = 0 = 1	Options for muon-induced electron-positron pair production. This reaction mechanism is not considered. This reaction mechanism is considered.
emumin	(D=200.0)	Minimum energy of muon-induced nuclear reaction [MeV]. The change of emumin also changes the reaction cross sections over the entire energy range.
emumax	(D=1.0e+6)	Maximum energy of muon-induced nuclear reaction [MeV].
npidk	(D=0) = 0 = 1	Treatment of negative-charged particles listed in Table 4.5 when they are below cut-off energies. Force to be absorbed by nucleus. Sets to decay.

For the nuclear resonance fluorescence considered in ipnint=2, polarization of incident photons can be specified using sx,sy,sz in the [source] section. The angular distribution of the scattered photons is determined with regard to the polarization.

If a particle with a decay channel (See Table 4.5) has an energy lower than the cut-off, the particle decays completely. Negative-charged pions (π^-) with the setting of npidk=0 are forced to be absorbed by nucleus. If π^- is not able to be absorbed in the program, the particle decays (to prevent an infinite calculation loop).

[23] Until ver. 2.30, this parameter was named ipngdr.

5.2.7 Model options (3)

Table 5.9: Parameters (9)

Parameter	Value	Explanation
imagnf	(D=0) = 0 = 1	Options for magnetic field. Without Magnetic field. With Magnetic field. Note that, to correctly describe the trajectory as a curve by tally, adjust the step length for the field by deltm.
ielctf	(D=0) = 0 = 1	Options for electromagnetic field. Without electromagnetic field. With electromagnetic field. Note that, to correctly describe the trajectory as a curve by tally, adjust the step length for the field by deltm.
andit	(D=0) = 0 = 1 = 2	Δ angular distribution for Bertini. 50 % isotropic, 50 % forward. All isotropic. All forward.
icxsni	(D=0) = 0 = 1 = 2	Options for inelastic scattering, elastic scattering, and total cross sections for nucleon-nucleus collisions. Pearlstein-Niita's formula. KUROTAMA model. Sato's formula.
icrhi	(D=2) = 0 = 1 = 2	Options for total cross section of nucleus-nucleus collisions. Shen formula. NASA formula. KUROTAMA model.
icrdm	(D=0) = 0 = 1	Options for total reaction cross sections for deuteron-induced reactions. Same as icrhi. MWO formula.
icxspi	(D=1) = 0 = 1	Options for total reaction cross sections for pion-induced reactions. Geometrical formula. Hashimoto's formula.
iidfs	(D=0) = 0 = 1	Options for neutron-induced fission. Use the nuclear data library. Neutron multiplicity ν and neutron energy spectrum are taken from the literature.
idwba	(D=0) = 0 = 1	Options for the Distorted Wave Born Approximation (DWBA) spectra. Without discrete DWBA spectra. With discrete DWBA spectra.

The KUROTAMA model covers a wide energy range of projectiles in nucleon-nucleus and nucleus-nucleus reaction cross sections. See the paper [24] for more details, and please cite the reference if your publication is based on the KUROTAMA model.

In the icrdm=1 setting, the MWO formula reproduces reaction cross sections of deuteron-induced reactions for incident energies below 1 GeV, and for target nuclei no lighter than C-12. See the reference [25] for more details; please cite this document in published results based the MWO model.

For the icxspi setting, the Hashimoto's formula has been used as the default model for calculating pion-induced reaction cross sections since version 2.86. This is an empirical formula that reproduces cross section experimental data better than the geometrical formula specified by icxspi=0.

If iidfs=1 is specified, the multiplicity ν and energy spectra of neutrons from neutron-induced fission taken from reference [26] are used instead of the nuclear library data for the following 18 nuclei: U-238, Pu-238, Pu-240, Pu-242, Cm-242, Cm-244, Cf-252, Th-232, U-232, U-233, U-234, U-235, U-236, Np-237, Pu-239, Pu-241, Am-241, and Bk-249.

[24] K. Iida, A. Kohama, and K. Oyamatsu, J. Phys. Soc. Japan 76, 044201 (2007), and L. Sihver *et al.*, Nucl. Instr. & Meth. B 334, 34-39 (2014).

[25] K. Minomo, K. Washiyama, and K. Ogata, J. Nucl. Sci. Technol. 54, 127-130 (2017).

[26] J. M. Verbeke, C. Hagmnn, and D. Wright, UCRL-AR-228518 (2014).

With the `idwba=1` setting, the discrete spectra calculated by the DWBA are added to the neutron and proton spectra obtained using other nuclear reaction models for the following reactions:

^{7}Li$(p, n)^{7}$Be reactions at 30–400 MeV
^{9}Be$(p, n)^{9}$B reactions at 10–50 MeV
^{6,7}Li$(d, n)^{7,8}$Be and ^{6,7}Li$(d, p)^{7,8}$Li reactions at 10–50 MeV
^{9}Be$(d, n)^{10}$B and ^{9}Be$(d, p)^{10}$Be reactions at 5–25 MeV
^{12,13}C$(d, n)^{13,14}$ and ^{12,13}C$(d, p)^{13,14}$C reactions at 10–50 MeV

See the reference [27] for more details.

[27] S. Hashimoto *et al.*, Nucl. Instr. & Meth. B 333, 27-41 (2014).

5.2.8 Model options (4) - stopping power

Table 5.10: Parameters (10)

Parameter	Value	Explanation
ndedx	(D=3)[28]	Options for dE/dx of charged particle and nucleus. Note that this option is not applicable for electrons and positrons.
	= 0	SPAR[29]for nucleus, NMTC for the rest of particle species.
	= 1	ATIMA[30]for nucleus and proton, NMTC for the rest of particle species.
	= 2	SPAR for nucleus, proton, pion, and muon, NMTC for the rest of particle species.
	= 3	ATIMA for all charged particles.
mdbatima	(D=500)	Maximum database size of ATIMA.
dbcutoff	(D=0.0)	Cut-off energy of ATIMA database [MeV/u].
ih2o	(D=-1)	Setting for the ionization potential of water in ATIMA. This setting is applicable for H_2O only.
	< 0	Use of the default 75 eV.
	> 0	User-defined value [eV].
ifixchg	(D=0)	Option for the charge number of source particles in ATIMA calculations.
	= 0	Effective charge. The charge number is equal to the Z number at high energies, and gradually decreases with decreasing energies.
	= 1	Fixed charge. The charge number is equal to the Z number as a constant, unless it is specified by the izst parameter in the [source] section.
irlet	(D=1)	Options for the method of considering δ-ray energy in calculating dE/dx for charged particle and nuclei.
	= 0	Conventional method (subtract the energy of sampled δ-ray from the unrestricted stopping power).
	= 1	Restricted LET. The cut-off energy of the delta rays is set in the [Delta Ray] section.
nspred	(D=0)	Options for Coulomb diffusion (angle straggling)[31]. Note that this option is not applicable for electrons and positrons.
	= 0	Without Coulomb diffusion.
	= 1	With Coulomb diffusion by the original NMTC model.
	= 2	(Recommended) With Coulomb diffusion by the Lynch's formula based on the Moliere theory[32].
	= 10	With Coulomb diffusion by ATIMA.
ascat1	(D=13.6)	S_2 parameter in the Lynch's formula for nspred=2.
ascat2	(D=0.038)	ϵ parameter in the Lynch's formula for nspred=2.
nedisp	(D=0)	Energy straggling option for charged particle and nuclei. Note that this option is not applicable for electrons and positrons.
	= 0	Without energy straggling.
	= 1	(Recommended) With Landau Vavilov energy straggling[33].
	= 10	With energy straggling for ATIMA.

[28] The default was ndedx=0 until PHITS ver. 2.00, and ndedx=2 from ver. 2.01 to ver. 2.85.

[29] T.M. Armstrong and K.C. Chandler, ORNL Report, ORNL-4869 (1973).

[30] About ATIMA, see the web site: https://web-docs.gsi.de/~weick/atima/

[31] When describing a scattering process on thin films less than 1cm, delt0 should be set to 1/10 of its thickness.

[32] $\sigma = S_2 \frac{\sqrt{X/X_0}}{p\beta}[1 + \epsilon \log_{10}(e) \log_e(X/X_0)]$: Eq.(4) in G.R. Lynch and O.I. Dahl, Nucl. Instrum. Methods Phys. Res, B 58, 6-10 (1991).

[33] About these models, see the web site: http://www.dnp.fmph.uniba.sk/cernlib/asdoc/geantold/H2GEANTPHYS332.html

From version 2.85, the algorithm for calculating stopping power with ATIMA in PHITS was improved and speeded up. In the old algorithm, calculation of range and stopping power was always performed every time when a new charged particle was produced or a transporting particle entered a new material. The new algorithm creates a database for each combination of transporting particle and material when a new particle-material combination appears, and look up the range and stopping power in the database the next time, instead of calling the high computational-cost routine ATIMA. As such, PHITS simulation using ATIMA is compatible with SPAR in terms of calculation time but with higher accuracy.

Starting from ver. 2.97, the setting of `ndedx=3` can be used for nuclei with the atomic number of $93 \leq Z \leq 97$ (from Np to Bk) [34]. Ionization potentials and density correction parameters needed in calculating nuclear stopping powers for those nuclei are adopted from the literature [35] , and SPAR is used in the calculation of electronic stopping power.

The setting of `ifixchg = 1` is not appropriate in most circumstances. The charge state of ions changes along their traveling paths in a material owing to attachment and disattachment of electrons. Thus, this setting only suits calculations of energy losses in a thin foil or gas.

[34] Prior to ver. 2.96, ATIMA cannot be used for nuclei with $Z > 92$ (U). For those cases, use SPAR by setting `ndedx=2`.

[35] Table I of R.M.Sternheimer, Atomic data and nuclear data tables, 30, 261-271 (1984).

5.2.9 Model options (5)

Table 5.11: Parameters (11)

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.
usrmgt	(D=1)	Options for user subroutine of time dependent magnetic field.
	= 1	usrmgt1.f, which includes Wobbler magnet, is used.
	= 2	usrmgt2.f, which includes Pulse magnet, is used.
usrelst	(D=1)	Options for [elastic option].
	= 1	usrelst1.f, for Bragg scattering, is used.
	= 2	usrelst2.f, a sample program, is used.
e-mode[36]	(D=0)	Options for event generator mode for neutrons below 20 MeV.
	= 0	Simple sampling mode (non event generator mode).
	= 1	Event generator mode Ver.1 (simple sampling + evaporation model).
	= 2	Event generator mode Ver.2 (Ver.1 + accurate charged particle yields + accurate neutron energy distribution).
em-emode	(D=20)	Maximum energy for e-mode [MeV].
ikerman	(D=0)	Control parameters for the use of kerma approximation for neutrons in [t-deposit].
	= 0	Automatic selection. Kerma approximation is not used when event generator mode is activated (e-mode ≥ 1).
	= 1	Kerma approximation for neutrons is not used.
	= 2	Kerma approximation for neutrons is used.
ikermap	(D=0)	Control parameters for the use of kerma approximation for photons in [t-deposit].
	= 0	Automatic selection. Kerma approximation is not used when EGS5 mode is activated (negs = 1), or when emin(12) and emin(13) are set below 10 MeV.
	= 1	Kerma approximation for photons is not used.
	= 2	Kerma approximation for photons is used.
ntrnore	(D=0)	Options for neutrino-induced reactions.
	= 0	This reaction mechanism is not taken into account.
	= 1	This reaction mechanism is taken into account.

gravx, gravy, gravz represent the directions of gravity; the gravitational force is effective for neutrons at energies below 1 eV. For example, if gravx=1, gravy=0, gravz=0, the direction of the gravitational force is negative along the x-axis.

The event generator mode (e-mode=1,2) is essential for calculation of event-by-event quantities, such as detector signals, residual recoiled nuclides, lineal energies, attributed to neutrons below 20 MeV. The simple sampling mode (e-mode=0) reproduces secondary neutron spectra given by the nuclear data library better than e-mode=1,2. Note that e-mode=0 does not consider the conservation of energy and momentum. See Sec. 5.2.22 for further discussion of the use.

By setting ntrnore $= 1$, following reactions are considered; ^{1}H$(\overline{\nu}_e, e^+)n$, ^{2}H$(\overline{\nu}_e, \overline{\nu}'_e)np$, ^{2}H$(\nu_e, e^-)2p$, ^{2}H$(\nu_e, \nu'_e)np$, and elastic scattering with electrons. Above 20 MeV, the cross sections of some reactions channels are extrapolated. Note that, the reaction channels above 150 MeV (e.g. charge-current reactions of μ-neutrinos) are not considered. Table 5.12 shows neutrino reaction systems considered in the current version of PHITS. In the case of that considered reaction channel, ∞ or values are shown as an energy upper-limit.

[36] Until ver. 2.73, file(14)=trxcrd.dat is required for e-mode=1,2.

Table 5.12: List of neutrino reaction systems considered and energies above which cross sections are extrapolated. ×: unimplemented, -: reaction is not existing.

Charge current

	e^-	^{1}H	^{2}H
ν_e	∞	-	20
$\overline{\nu}_e$	-	150	20
ν_μ	×	-	×
$\overline{\nu}_\mu$	-	×	×
ν_τ	×	-	×
$\overline{\nu}_\tau$	-	×	×

Neutral current

	e^-	^{1}H	^{2}H
ν_e	∞	×	20
$\overline{\nu}_e$	∞	×	20
ν_μ	∞	×	20
$\overline{\nu}_\mu$	∞	×	20
ν_τ	∞	×	20
$\overline{\nu}_\tau$	∞	×	20

5.2.10 Output options (1)

Table 5.13: Parameters (12)

Parameter	Value	Explanation
`infout`	`(D=7)`	Options to specify output information in `file(6)`. The notes followed by this table give the meanings of the category codes.
	`= 0`	I
	`= 1`	I, II
	`= 2`	I, III
	`= 3`	I, IV
	`= 4`	I, II, III
	`= 5`	I, II, IV
	`= 6`	I, III, IV
	`= 7`	I, II, III, IV, VI
	`= 8`	I, II, III, IV, V, VI
`nrecover`	`(D=0)`	Number of warning messages output when the recovery of lost particles succeeds.

The `infout` parameter controls output information in the summary file, `file(6)`. Using `infout`, the output information can be selected by the user. `infout` information is divided into the following six categories:

I Basic information – LOGO (excluding the information on PHITS developers), calculation process, source, geometry error, random seed, and CPU summary.

II Input echo.

III Information on memory usage and batch.

IV Information on transport particles.

V Detailed information – variance reduction, number of scattered particles for each region, information for each material.

VI Information on PHITS developers.

The information is output in the following order:

(1) LOGO (Category I). Version of PHITS, developers information (Category VI), job title, and starting time.

(2) Echo of input file (Category II).

(3) Information on memory usage and composition for geometry, material, tally, bank, etc. (Category III).

(4) Information on batch (Category III)
Number of histories by batch and CPU time.

(5) Information on calculation process (ncol) (Category I).
Number of geometry boundary crossings, reactions, terminations by energy cut-off, etc.

(6) Information on variance reduction (Category V).
Particle weights changed by importance, weight window, and forced collision options.

(7) Number of scattered particles for each region (Category V).

(8) Number of scattered particles for each material (Category V).

(9) Number of transport, produced, stop, and leakage particles (Category IV).

(10) Number of source particles and their weights (Category I).

(11) Geometry error (Category I).
Number of lost particles and their types.

(12) Random seed (Category I).
Initial random seed and next initial random seed.

(13) CPU summary (Category I).
Total computation time and number of calls for each calculation process. The meanings of each term are explained as follows:

- total cpu time: total computation time.
- transport: time for particle transport.
- set data: time for setting input parameters.
- analysis: number of data processing events.
- nevap: number of evaporations.
- dexgam: number of de-excitations.
- nreac: number of atomic and nuclear reactions.
- dklos: number of particle decays.
- hydro: number of nucleon-nucleon scatterings.
- n-data: number of runs using neutron data library.
- h-data: number of runs using proton data library.
- p-data: number of runs using photon data library.
- e-data: number of runs using electron data library.
- p-egs5: number of photon interactions with EGS5.
- e-egs5: number of electron interactions with EGS5.
- photonucl: number of photo-nuclear reactions.
- muon: number of muon-induced nuclear reactions.
- elast: number of elastic scatterings.
- ncasc: number of runs using nuclear reaction models.
- bertini: number of runs using Bertini model.
- isobar: number of runs using isobar model.
- JAM: number of runs using JAM model.
- QMD: number of runs using JQMD or JQMD-2.0 models.
- JAMQMD: number of runs using JAMQMD model.
- INCL: number of runs using INCL model.
- INCELF: number of runs using INC-ELF model.

Note that when `icput=0` by default, output information does include the computation time (except for ‘total cpu time’), ‘transport’, nor ‘set data.’
When `infout=8`, information on the calculations by each nuclear reaction model is output in detail.

- count: number of runs using the models with each incident particle.
- real: number of successes for model calculation.
- %: probability of model calculation.

5.2.11 Output options (2)

Table 5.14: Parameters (13)

Parameter	Value	Explanation
`incut`	(D=0)	Neutron output options below the cut-off energy.
	= 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 the cut-off energy.
	= 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 the cut-off energy.
	= 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).
`dumpall`	(D=0)	`dumpall` option. Activate and use with `icntl=12`. See 5.2.21 for more details.
	= 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).

In the setting of `inpara, igpara, ippara, idpara`, “/wk/” and “/uname/” are the default directory name and a user-name read in from the environment variable `LOGNAME`, respectively. In parallel computing, files corresponding to each processor element (PE) are created for writing the output or the dumped data. If `inpara, igpara, ippara`, or `idpara` is set to 0 or 1, a file is made in the directory named by /wk/uname/ on each of the nodes. If `inpara, igpara, ippara`, or `idpara` is set to 1 or 3, the PE number is added at the end of the filename. Each PE writes down its calculation result only on its corresponding file.

5.2.12 Output options (3)

Table 5.15: Parameters (14)

Parameter	Value	Explanation
itall	(D=0)	Options for tally output after every batch.
	= -1	No output.
	= 0	Output only numerical data files.
	= 1	Output both numerical data and image (*.eps) files[37].
	= 2	Output to different files. (file name = specified file name + batch number))
	= 3	Output transition of tally results and these statistical errors as a function of the history number (or batch number). These values are output into a file '*_StD' where '*' indicates the file name specified in the tally section. This function is available only for [t-track] and [t-point] in the current version of PHITS.
iMeVperu	(D=0)	Options for the unit of energy in tally output.
	=0	Output in MeV.
	=1	Output in MeV/u (MeV per nucleons) for nuclei. The unit automatically changes to MeV for the rest of the particle species.
itstep	(D=1)	Options for the tallying timing when the tracking particle is in a field, which constantly changes the particle's momentum, such as a magnetic, electromagnetic, or gravity field.
	= 0	Tally at reaction or surface cross (normal).
	= 1	Tally at each step of the transport.
imout	(D=0)	Options for the output format of the [material] section.
	= 0	e.g., mat[12], 208Pb.33c.
	= 1	e.g., mat[12], Pb-208.33c.
	= 2	e.g., m12, 82208.33c (MCNP style).
jmout	(D=0)	Options for the output format of material density in [material].
	= 0	No conversion.
	= 1	Conversion into particle density.
	= 2	Conversion into mass ratio (%).
kmout	(D=0)	Options of nuclear data information.
	= 0	No display.
	= 1	Writing in input echo.[38]

When iMeVperu=1, the unit of nucleus energy is changed from [MeV] to [MeV/u] in outputs of [t-track], [t-cross], [t-point], [t-product], [t-time], [t-interact]. For example, in the case of [t-track] with unit = 2, quantities in units of [1/cm^2/MeV/source] are converted to those in [1/cm^2/(MeV/u)/source].

[t-track] uses deltm as the step length to describe particle trajectories in fields that continuouly change particles' momenta, such as a magnetic field. To describe the trajectory curve more smoothly, adjust the setting of deltm.

[37] Note that vtk files are output with both itall=0,1.

[38] From Ver. 2.86, the dmax of each nucleus is also output.

5.2.13 Output options (4)

Table 5.16: Parameters (15)

Parameter	Value	Explanation
matadd	(D=1)	Options when different densities are defined for a given material number in the [cell] section.
	= 0	Use the same material number.
	= 1	Use a new material number.
natural	(D=1)	Options for using the naturally occurring isotopes for nuclear cross section data, and for writing their ratios in the input echo, when a given element is defined without the mass number in [material].
	= 0	Use nuclear cross section data for natural elements, such as ^{nat}Fe and ^{nat}Pb, in the nuclear data library.[39] Isotopes are not output in file(6).
	= 1	Assign all the naturally occurring isotopes to the given element. Isotopes are not output in file(6).
	= 2	Assign all the naturally occurring isotopes to the given element. Isotopes are output in file(6).
iggcm	(D=0)	Option of GG warnings.
	= 0	No echo.
	= 1	In input echo.
icput	(D=0)	Options for recording and outputting CPU time for each calculation process.
	= 0	Not recorded nor output.
	= 1	Recorded and output.
ipara	(D=0)	Parameter display options.
	= 0	Only described parameters.
	= 1	All parameters.
nwsors	(D=0)	Write down the information on nwsors source particles in file(6).

When two (or more) cells are defined with the same material number but differing densities, with the default setting of matadd=1, such an material will be given a new material number as it is redefined by the other density. If matadd=0 is set, the cells will have the same material number. Note that this setting is invalid in the [mat name color], in which other given material numbers are output as a warning message in the first part of file(6) (D=phits.out).

To save calculation time, CPU time for each calculation process, such as transport, set data, analysis, etc., is not recorded and output by default. To obtain such information, set icput=1.

Setting ipara=1 enables confirmation of all parameters in the PHITS code.

[39] An error message occurs if such a natural element does not exist in the data library.

5.2.14 Voxel and tetrahedron geometry options

Table 5.17: Parameters (16)

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).
itetvol	(D=0)	Options to compute volume of tetrahedron geometry for LAT=3.
	= 0	The volume will not be computed.
	= 1	The volume will be automatically computed and used in tallies, etc.
itetra	(D=0)	Read and write tetrahedron geometry data in binary. Binary file must be reconstructed by setting itetra=2 when ntetsurf and/or ntetelem are changed.
	= 0	Not using the data file.
	= 1	Read from tetrahedron geometry data in binary “Tetra.bin” (fixed name).
	= 2	Write down tetrahedron geometry data in binary “Tetra.bin” (fixed name).
ntetsurf	(D=100)	Number of outer surfaces of tetrahedron geometry allowed for each sub-space of the container box.
ntetelem	(D=200)	Number of elements of tetrahedron geometry allowed for each sub-space of the container box.
itetauto	(D=0)	Options to automatically create cells corresponding to universes of tetrahedron geometry.
	= 0	No cell is automatically created.
	= 1	The cells will be automatically created. The text file (.txt) is required for the TetGen format.

The ivoxel parameter can be used to accelerate calcutions with voxel geometry. When performing PHITS calculation with ivoxel=2, voxel data are output in file(18) in binary format, and the calculation is then stopped (until ver. 2.30, calculation was continued). In the next calculation with ivoxel=1, the necessary information is read from file(18) created in the previous calculation with ivoxel=2, so the data output process is omitted and the calculation time is shortened.

Using the option of itetvol=1, when the tetrahedron geometry is treated with LAT=3, the volume of the universes appearing in tetrahedron geometry can be automatically computed. However, the volume is computed by simply summing up the volumes of individual tetrahedrons, which might produce incorrect values in cases in which tetrahedron geometry is clipped out through the use of nest structure of “universe”and “fill”. The computed value with the option of itetvol=1 is given priority over values specified in the [volume] or other sections.

When tetrahedron geometry (LAT=3) is used in PHITS calculation, necessary information for transport calculation is generated by reading the tetrahedron geometry file prior to the actual transport calculation. This process becomes computationally heavy as the number of elements in the tetrahedron geometry increases. In such case, the itetra parameter can be used to reduce the computational time. Performing PHITS calculation with itetra=2 will have the processed tetrahedron geometry data recorded in the binary file (“Tetra.bin”) followed by the calculation being stopped. In the next calculation with itetra=1, the necessary information is read from the Tetra.bin file, and the data generation process is omitted to reduce the calculation time.

The ntetsurf and ntetelem parameters are associated with the two algorithms sub-dividing the box containing the tetrahedron geometry for PHITS calculations with (LAT=3), which reduce the computational time by limiting the number of elements belonging to each sub-space. One algorithm finds a surface of the tetrahedron geometry when a particle enters the geometry. The container box is sub-divided so that the number of outer surfaces at each sub-space is less than ntetsurf. The other algorithm finds an element of tetrahedron geometry when a particle is initiated inside the geometry. The box containing all tetrahedron elements is subdivided so that the number of elements belonging to each sub-space is less than ntetelem.

Using the option of itetauto=1, cells required by each tetrahedron with specfied universe in tetrahedron geometry are created automatically. The cost manually creating the cells can be reduced with this option when the number of universes contained in tetrahedron geometry increase. This option is especially helpful when the tetrahedron geometry is imported with the NASTRAN bulk data format. See 5.6.5 for more details.

5.2.15 About geometrical errors

Table 5.18: Parameters (17)

Parameter	Value	Explanation
`nlost`	`(D=10)`	Acceptable value against lost particle (per 1 PE).
`igerr`	`(D=0)`	Number of recoveries from region error.

`igchk`	`(D=0)`
	`= 0`
	`= 1`
		Options for region check.
		No region check.
		Set the flight mesh to `deltb` after region-crossing, and check region.

`deltb`	`(D=1.e-9)`	Distance of flight mesh [cm] after region-crossing if `igchk=1`. `(deltb is also the distance from the region boundary to where the particle is created by importance and forced collision.)`

`idelt`	`(D=1)`
	`= 0`
	`= 1`
		Options for minimum flight mesh.
		Use values of `deltm` and `deltc` as flight mesh maxima.
		Maximum flight mesh is `deltm` and `deltc` divided by the density in the region.

`deltm`	`(D=20.12345)`	Maximum flight mesh [cm]. This parameter is linked with `idelt`.
`deltc`	`(D=2.012345)`	Maximum flight mesh [cm] for charged particles if `nedisp=1,10` (not applicable for electrons and positrons). This parameter is linked with `idelt`.
`delt0`	`(D=0.1)`	Minimum flight mesh [cm] if `nspred=1,2,10`. Note that, when describing a scattering process on thin films less than 1 cm, `delt0` should be set to 1/10 of its thickness.
`deltg`	`(D=1.012345)`	Maximum flight mesh [cm] in a magnetic field for charged particles (excluding electrons and positrons).
`deltt`	`(D=1.0)`	Maximum flight time [msec] in a time-dependent magnetic field.

If `igchk` is set to 1, after a particle crosses the boundary of a region, the particle's location is artificially moved a short distance away from the boundary surface along the forward direction in the PHITS code, and this short distance is defined as "flight mesh."

The purpose of the use of flight mesh is to more accurately determine which region the particle belongs to, for prevention of misjudments due to insufficient significant digits. This function is important in particular for large scale simulations that use curved surfaces.

From ver. 2.80, `idelt=1` has become the default setting of PHITS, which sets the quotients of `deltm` and `deltc` divided by the region's density (in units of g/cm^3) as the flight mesh maxima. This setting is useful in for shortening the particle transport calculation time in air with a spatial scale over several hundred meters.

5.2.16 Input-output file name

Table 5.19: Parameters (18)

Parameter	Value	Explanation
file(1)	(D=c:/phits)	PHITS installation folder name. If this parameter is set properly, there is no need to specify the names of other input files, i.e. file(7, 20, 21, 24, 25, and 26), unless the folder structure of PHITS have not been changed. If these parameters are directly specified in the input file, the specified parameters overwrite the setting.
file(6)	(D=phits.out)	Summary output file name.
file(7)	(D=file(1)/data/xsdir.jnd)	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(15)	(D=dumpall.dat)	dump file name for dumpall=1 option.
file(18)	(D=voxel.bin)	File name when using ivoxel=1, 2.
file(20)	(D=file(1)/XS/egs/)	Directory containing the library data for EGS5.
file(21)	(D=file(1)/dchain-sp/data/)	Directory containing the library data for DCHAIN.
file(22)	(D=batch.out)	Name of the file to output information on current batch.
file(23)	(D=pegs5)	Output file name for PEGS5.
file(24)	(D=file(1)/data)	Directory containing RIsource.dat (DECDC database).
file(25)	(D=file(1)/XS/tra)	Directory containing cross section database used for track-structure simulation.
file(26)	(D=file(1)/data/multiplier)	Directory containing the pre-defined [multiplier] sections.
file(27)	(D=file(1)/XS/yield/)	Directory containing the user-defined activation cross section data. The last "/" of the path is needed.
file(28)	(D=file(1)/data/aama.dat)	Input file name for the MUONIC ATOM CASCADE program to use the user-defined parameters.

file(7) must be written with the full pathname. From ver. 2.74, the file(14) parameter does not have to be specified in the PHITS input file even when setting |igamma|≥1.

5.2.17 Others

Table 5.20: Parameters (19)

Parameter	Value	Explanation
`inucr`	(D=1) = 1 = 2 = 3 = 4 = 5 = 6	Nuclear reaction calculation options. Double differential cross section calculation. Total, elastic, non-elastic cross section calculation. Non-elastic cross section calculation. Angular distribution of elastic scattering. pp, np, πp cross section output. pp, np, πp cross section calculation.

`idam(i)`
`rdam(i)`
	integer
	real*8
	i = 1-100
		User-defined integer variable.
		User-defined real variable.
		These values can be used in the PHITS code by
		`common /userp/ idam(100), rdam(100)`

`icommat`	(D=0) = 0 = 1	Option for comment marks in `[material]` section. `c` cannot be used as comment marks. `c` can be used as comment marks.

The nuclear reaction calculation mode with `inucr` is corresponding to `icntl=1`, which is currently under development.

From version 2.89, in the default setting, ‘c’ cannot be used as comment marks in `[material]` section. To use ‘c’ as comment marks in the section, set `icommat=1`. Note that the `[parameters]` section with `icommat=1` should be written above (before) the `[material]` section including ‘c.’

5.2.18 Physical parameters for low energy neutrons

The following parameters are used for neutrons with energies less than 20 MeV.

Table 5.21: Parameters (20)

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.
`dnb`	(D=-1) =-1 = 0 > 0	Number of delayed neutron produced by fission. Natural sampling. No delayed neutrons. Specify number of delayed neutrons.
`nonu`	(D=1) =0 =1	Options for fission neutrons Turn off fission: fission is treated as simple capture. Normal fission producing neutrons.
`isaba`	(D=0) = 0 = 1	Option for $S(\alpha,\beta)$. No interpolation. Perform interpolation of the data of $S(\alpha,\beta)$, which should be specified for calculations with thin targets in which particle scattering rarely occurs.

5.2.19 Physical parameters for photon and electron transport based on the PHITS original model

Table 5.22: Parameters (21)

Parameter	Value	Explanation
emcpf	(D=100)	Maximum energy in the detailed photon model [MeV].
ides	(D=1)	Electron creation options by photon.[40]
	= 0	Create electron or bremsstrahlung photon.
	= 1	Do not create electron.
nocoh	(D=0)	Photon coherent scattering options.
	= 0	With coherent scattering.
	= 1	Without coherent scattering.
iphot	(D=0)	Photon creation options by electron.
	= 0	Create photon.
	= 1	Do not create photon.
ibad	(D=0)	Angular distribution option for bremsstrahlung.
	= 0	Full bremsstrahlung tabular angular distribution.
	= 1	Simple bremsstrahlung angular distribution approximation.
istrg	(D=0)	Energy straggling options for electrons.
	= 0	Sampled straggling for electron energy loss.
	= 1	Expected value straggling for electron energy loss.
bnum	(D=1)	Options for bremsstrahlung photon.
	= 0	Do not create bremsstrahlung photon.
	> 0	Number of analog bremsstrahlung photons.
xnum	(D=1)	Options for X-ray photon.
	= 0	Not create X-ray photon.
	> 0	Number of analog X-ray photons.
enum	(D=1)	Options for secondary electron.
	= 0	Do not create secondary electrons.
	> 0	Number of analog secondary electrons.
numb	(D=0)	Options for bremsstrahlung production process.
	= 0	Nominal bremsstrahlung production.
	> 0	Produce bremsstrahlung at each substep.
rnok	(D=1)	Options for knock-on electron.
	= 0	Do not create knock-on electrons.
	> 0	Number of knock-on electrons.

[40] ides is automatically set to 0 when either emin(12) or emin(13) or both is defined in the input file from ver. 2.96.

5.2.20 Parameters for EGS5

Table 5.23: Parameters (22)

Parameter	Value	Explanation
`ipegs`	(D=0)	Options for controlling PEGS5 (invalid when `negs`≠ 1).
	= -1	Output PEGS5.inp: Yes, Execute PEGS5: Yes, Execute PHITS: No.
	= 0	Output PEGS5.inp: Yes, Execute PEGS5: Yes, Execute PHITS: Yes.
	= 1	Output PEGS5.inp: No, Execute PEGS5: Yes, Execute PHITS: Yes.
	= 2	Output PEGS5.inp: No, Execute PEGS5: No, Execute PHITS: Yes.
`imsegs`	(D=1)	Options for treating multiple scatterings in EGS5.
	= 0	Original EGS5 method. With this option, the starting scattering strength is used only when the electron is generated as a source particle. After several steps, the maximum scattering strength is used.
	= 1	Original PHITS-EGS5 method. Use starting scattering strength whenever an electron enters a new material. This option enables transporting electrons in thin foils without changing the `chard` parameter; the computational time becomes slightly longer.
`iegsout`	(D=0)	Options for controlling EGS5 output files.
	= 0	Delete all EGS5 output files when PHITS calculation is finished. Some files remain when the user self-terminates the PHITS calculation.
	= 1	Keep `egs5.inp, pegs5.dat` and `pegs5.msfit` and delete the others.
	= 2	Keep all EGS5 output files. See EGS5 manual[41] for details on each file.
`iegsrand`	(D=-1)	Options for random number used in EGS5. Setting parameter to 0 or to a positive value hides the MPI parallelization and the restart-calculation function.
	< 0	Use random number generated by PHITS.
	= 0	Use random number generated by EGS5 with the default initial random seed (314159265).
	> 0	Use random number generated by EGS5 with the initial random seed = `iegsrand`.

PEGS is the prepocessor for EGS. It contains a set of fortran subprograms, which negerate material data for the use of EGS5. It is needed to change the parameter `ipegs` in order to change the input file of PEGS5 (`pegs5.inp`). Note that, to retain PEGS5-related files after executing PHITS, `iegsout` automatically becomes 1 (or greater) when `ipegs` is set ≥ 1.

When using EGS5, the default values of the minimum and maximum energies for electron, positron, and photon transports are changed to `emin(12,13)=0.1, emin(14)=0.001, and dmax(12-14)=1,000.0`, respectively. These values can also be changed by explicitly setting the related parameters. Note that the default values of several EGS5 parameters differ from those employed in the original EGS5 code.

From ver. 2.76, photo-nuclear reactions can be considered in EGS5. Thus, input files with `negs=1` and `ipnint=1` produce results that differ from those calculated by the previous version of PHITS.

[41] H. Hirayama *et al.*, SLAC-R-730 (2005) and KEK Report 2005-8 (2005).

Table 5.24: Parameters (23)

Parameter	Value	Explanation
iedgfl	(D=1) = 0 = 1	Option for K and L-edge fluorescent photons. Do not explicitly treat K and L-edge fluorescent photons. Explicitly treat K and L-edge fluorescent photons.
iauger	(D=1) = 0 = 1	Option for K and L-edge Auger electrons. Do not explicitly treat K and L-edge Auger electrons. Explicitly treat K and L-edge Auger electrons.
iannih	(D=0) = 0 = 1	Option for the production of annihilation photons (only for proj = photon) Consider the production of annihilation photons. Ignore the production of annihilation photons. This option should be selected when you explicitly consider the production of positrons by defining another <source> with proj = positron, otherwise annihilation photons will be counted twice.
iraylr	(D=1) = 0 = 1	Option for coherent (Rayleigh) scattering. Do not consider coherent scattering. Consider coherent scattering.
lpolar	(D=0) = 0 = 1	Option for linearly polarized photon scattering (not available in the PHITS code at this moment). Do not consider linearly polarized photon scattering. Consider linearly polarized photon scattering.
iunrst	(D=0) = 1 = 2 = 3 = 4 = 5 = 6 = 7	Option for outputting stopping powers by PEGS5. When this parameter is changed, ipegs must be set to −1 to not to execute PHITS calculations, and to prevent obtaining incorrect results. Unrestricted collisional stopping power only. Unrestricted collisional and radiative stopping power. Unrestricted collisional and restricted radiative stopping power. Restricted collisional and unrestricted radiative stopping power. Unrestricted radiative stopping power only. Restricted radiative stopping power only. Restricted collisional stopping power only.
chard	(D=0.1)	Array of ‘characteristic dimensions,’ or representative size (in units of cm) of each material. This parameter controls the electron step-size in each material. The value must be decreased when simulating electron scattering in thin foils by setting imsegs=0. When a negative value is set for this parameter, the actual chard values used in EGS5 are adjusted to their absolute values divided by the material density in g/cm^3. This option is useful for simulating electrons in huge objects (of km order).
epstfl	(D=0) = 0 = 1	Option for using density correction factors contained in EPSTAR. Do not use EPSTAR for density correction. Use EPSTAR for density correction. If this option is selected, a text file named “epstar.dat” must be created in the same folder containing the PHITS input file. The data contained in “\XS\density_corrections” must be copied and pasted in the order of materials specified in the [material] section. This can be directly changed for each material by editing “pegs5.inp.”

Table 5.25: Parameters (24)

Parameter	Value	Explanation
incohr	(D=1)	Option for incoherent scattering function.
	= 0	Do not use incoherent scattering function for Compton scattering.
	= 1	Use incoherent scattering function for Compton scattering (ibound is automatically set to 1).
iprofr	(D=1)	Option for Doppler broadening of Compton scattering energies.
	= 0	Do not consider Doppler broadening.
	= 1	Consider Doppler broadening (incohr and ibound are both automatically set to 1).
impacr	(D=1)	Option for electron-impact ionization (EII).
	= 0	Do not consider EII.
	= 1	Consider EII.
ieispl	(D=0)	Option for split of X-rays generated by electron-impact ionization.
	= 0	No splitting.
	= 1	Splitting.
neispl	(D=0)	Number of electron-impact ionization X-rays for splitting when ieispl=1.
ibrdst	(D=1)	Option for determination of bremsstrahlung polar angle.
	= 0	Angle given by m/E.
	= 1	Sampling.
iprdst	(D=1)	Option for determination of polar angles of pair electrons.
	= 0	Angle given by m/k.
	= 1	Sampling.
iphter	(D=1)	Option for determining angular distributions of photoelectrons.
	= 0	Emission in the direction of incident photon.
	= 1	Sampling.
ibound	(D=1)	Option for Compton cross section.
	= 0	Use free Compton cross section.
	= 1	Use bound total Compton cross section.
iaprim	(D=1)	Option for correcting bremsstrahlung cross section.
	= 0	Use Motz *et al.* empirical.
	= 1	Normalize integrated cross section to ICRU-37 radiative stopping power.
	= 2	No correction.

5.2.21 Dumpall option

With the setting of `icntl=12`, PHITS reads the detailed information of a completed transport calculation from a `dumpall` file created beforehand by using the `dumpall` option, and reproduces all of the transport result event by event. As such, the input file except the tally sections must be identical as used in the previous calculation when the `dumpall` option is activated. `maxcas` and `maxbch` cannot be changed either, because they are read from the dump file.
This can be a very useful function in order to calculate different tallies which were not used in the previous calculation. However, please be careful that the calculation with the `dumpall` option may create a huge dump file.

The dumped data written in ascii can be used on the computers, but the data written in binary cannot. The data sequence and meaning are given in the following.

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

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 the charge state of particles.

(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
13	:	Photon reactions by EGS5
14	:	Electron reactions by EGS5

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 an 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	:	x-component of unit vector of momentum
= 2	:	y-component of unit vector of momentum
= 3	:	z-component of unit vector of momentum
= 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)

5.2.22 Event Generator Mode

For Event Generator mode [42], dmax(2) should be defined appropriately using information from the nuclear data library. In the special statistical decay model, detailed information on the level structure near the ground state for particle and photon emission is used, which requires setting igamma=1-3. The special statistical decay model has been developed based on GEM; to use it, the operator should specify nevap=3. The related parameters are automatically set by specifying e-mode=1 or 2 (unless explicitly specified, igamma=2 is selected). For consistency reason, the combination igamma=1 and e-mode=1 is also supported.

Event generator mode Ver.2 is the improved version of the legacy version (i.e., event generator mode Ver.1). In reactions emitting multiple neutrons, the previous event generator mode sampled the first ejectile neutron from the cross-section data and emission of the subsequent particles was simulated using the statistical decay model. By contrast, ver.2 samples all ejectile neutrons from the cross-section data and the statistical decay model is used merely to simulate prompt gamma-ray production. For capture reactions, the previous event generator mode assumed that the target nucleus absorbs incident neutrons, and particle emission was simulated by the statistical decay model. In this updated version, ejectile particle species are fully determined by the statistical decay model. While ver.2 selects the ejectile particle species depending on the reaction channel (i.e., in the (n, α) reaction, emission of only one alpha particle and gamma-rays is allowed).

Under Event Generator mode, the following new observables, which cannot be detected otherwise, are obtained:

(1) The deposition energy distribution by the low-energy neutrons below dmax(2) without the kerma approximation is available in the [t-deposit] tally by considering contributions from the secondary charged particles. When e-mode=0, the deposition energy by the low-energy neutrons in [t-deposit] is calculated with the kerma approximation [43].

(2) In [t-yield] and [t-product], the yield and product quantities can be tallied below dmax(2).

(3) DPA values are obtained even for neutrons of energies below dmax(2).

[42] Y. Iwamoto *et al.*, International Conference on Nuclear Data for Science and Technology 2007, DOI: 10.1051/ndata:07417; K. Niita *et al.*, International Conference on Nuclear Data for Science and Technology 2007, DOI: 10.1051/ndata:07398; Y. Iwamoto *et al.*, Prog. Nucl. Sci. Technol. 2, 931-935 (2011).

[43] After ver. 3.05. Before ver. 3.04, the [t-deposit] tally cannot calculate the deposition energy with the kerma approximation.

5.3 [Source] section

The source information can be set in this section. The source type is specified by the number `s-type=N` based on its geometry as in Table 5.26.

The energy of the source particle can be given by `e0` for mono-energy or `e-type` for energy spectrum [44].

Table 5.26: Source type

source type	explanation
`s-type = 1, (4)`	Cylinder (or circle, pencil)
`s-type = 2, (5)`	Rectangular solid (or rectangle)
`s-type = 3, (6)`	Gaussian (x, y, z independent)
`s-type = 7, (8)`	Generic parabola (x, y, z independent)
`s-type = 9, (10)`	Sphere or spherical shell
`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, (14)`	Gaussian (*xy*x plane)
`s-type = 15, (16)`	Generic parabola (*xy* plane)
`s-type = 17`	Reading dump file
`s-type = 18, (19)`	Cone
`s-type = 20, (21)`	Triangle prism
`s-type = 22, (23)`	*xyz*-mesh distribution
`s-type = 24`	Tetra-mesh source
`s-type = 26`	Surface definition source
`s-type = 100`	User definition source Edit the usrsors.f and compile PHITS.

In Table 5.26, numbers in the brackets were used until ver. 2.94. If both `e0` and `e-type` are defined from ver. 2.95, the energy is decided according to the previous procedure of `s-type`. For example, `e0` is used when `s-type=1`, and the definition of `e-type` is used when `s-type=4`.

[44] Until ver. 2.94, `s-type` had to be used depending on the definition of the energy, e.g., `s-type=1` with `e0` defined the cylinder source of the mono-energy, or `s-type=4` with `e-type` defined that of the energy spectrum.

5.3.1 <Source> : Multi-source

Using this multi-source function, multiple sources with different source definitions, such as different particle species, geometries, or energy distributions, can be specified. Each source definition begins with `<source>=`*number*, which specifies the relative weight of such type of source. `totfact=` defines a global scaling factor.

Table 5.27: Multi-source

parameter	explanation
`<source> =`	The relative weight of such source definition. This is necessary when multiple sources are defined.
`totfact =`	(D=1) Global scaling factor.
`iscorr =`	(D=0) Multi-source correlation option. 0: Uncorrelated multi-source. 1: Correlated multi-source. In this case, source particles from each multi-source definition are generated as an event. This option is useful for estimating detector response by nuclear reactions producing several secondary particles. Note that the locations of all sources generated in an event are not the same, unlike the case of `iscorr = 2,3`. 2: Correlated multi-source. In this case, the locations of all sources generated in an event are the same. This mode is useful for simulating a nuclear reaction occurring at an arbitrary location. 3: Correlated multi-source. In this case, the locations of all sources generated in an event are the same, and the direction of the second source is opposite to that of the first. This mode is useful for simulating a particle decaying into two particles at an arbitrary location.

Tally results in PHITS calculations are normalized to per source particle, or per unit weight in a more strict sense. For the `totfact` setting, if it is given a positive value, number of source particles are generated based on the intensity ratio of each source definition; if negative, same number of source particles are generated for each source definition, but their weights are adjusted to match the intensity ratio of all source definitions. If one source definition has a much lower intensity than the others, yet such source definition contributes significantly to the tally results, it is recommended to set `totfact<0` to improve calculation efficiency.

For `iscorr = 1-3`, the multiplicity of each source must be specified as an integer value representing its `<source>` parameter, and the sum of the multiplicities must be specified as the `totfact` parameter. For example, to simulate $(X, 1p2n)$ reaction as an event, set `<source>=1` and 2 for proton and neutron sources, respectively, and `totfact=3`.

5.3.2 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=***), it can be omitted. The energies of d, t, α, and nuclei are specified by [MeV/nucleon]. Please see sections 5.3.19, 5.3.20, 5.3.21 for the definitions of energy, angular, and time distributions, respectively for source description.

Table 5.28: Common source parameters (1)

parameter	explanation
`proj =`	Projectile: see Table4.4 for specification.

`sx =`
`sy =`
`sz =`
	(D=0) x-component of spin.
	(D=0) y-component of spin.
	(D=0) z-component of spin.

`reg =`	(D=all) The source region can be restricted to the overlap(s) between the regions defined by `s-type` and those specified by this parameter. The format is `reg = { 1 - 5 } 10 34`. The lattice and universe frame can be used as `reg = ( 6 < 10[1 0 0] < u=3 )`. See the section of tally region specification for more details on the syntax. Note that this parameter cannot be written in the line right above `<source>` = if multi-source sub-sections are defined; otherwise it will cause a syntax error.
`ntmax =`	(D=1000) Maximum number of re-sampling the source location from the region specified by `s-type` when `reg` is specified.
`trcl =`	(D=none) Transform number, or definition of transform.
`wgt =`	(D=1.0) Weight of source particles.
`factor =`	(D=1.0) Scaling factor fot tally results; all tally results are multiplied by the value of `factor`. When `factor` and `totfact` are both defined, `totfact` × `factor` is the actual scaling factor. However, when multi-source is defined, use `totfact` only; do not use this parameter for scaling.
`izst =`	(D=charge of particle species specified by `proj=`) Charge state of source particle. This value influences the particle motions only in the magnetic and electro-magnetic fields defined in `[magnetic field]` and `[electro magnetic field]` sections. In addition, ATIMA with the fixed charge mode (`ifixchg=1`) calculates the stopping power based on this charge state. The charge number defined with `izst` does not change while the particle travels. Note that this parameter is only valid for source particles, i.e., secondary particles produced from nuclear reactions carry the charges of their atomic numbers.
`cnt(i)=`	(D=0) Initial value of counter `i (i=1-3)` of source particle. You can distinguish the contributions from each multi-source definition by assigning different values to their initial counter values.

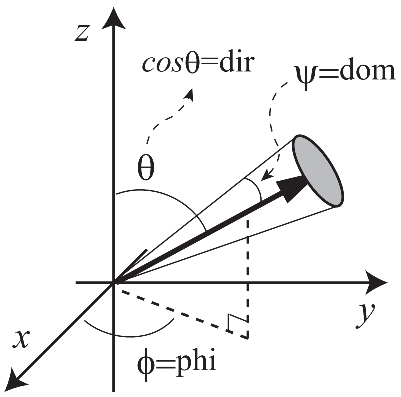


Figure 5.2: Source direction and parameters `dir, phi, dom`.

The projectile direction is specified by three parameters: `dir, phi,` and `dom`. The relation between these is shown in Fig. 5.2, in which the direction is noted as a thick arrow. `dir` is the direction cosine relative to the *z* axis. `phi` is the azimuthal angle from the *x* axis and is given in degrees; if not set by the user, the value of the azimuthal cosine is selected randomly. Using the parameter `dom` spreads the direction determined by `dir` and `phi` by the solid angle $2\pi(1 - \cos\psi)$, where ψ = `dom` (in degrees). In the PHITS calculation, the angle is selected randomly within the solid angle.

By setting `dir=all`, the direction of the source beam becomes isotropic. To use a specific angular distribution, a subsection in the distribution based on numerical data or an analytic function starting from `a-type` is required.

In `s-type=9`, the definition of `dir` differs. In `s-type=11` and `12`, `dir` can only bet set = −1 or 1.

To transform coordinates, the `trcl` parameter, which specifies the transform number or the definition of the transformation itself, can be used. The relation of `wgt` and `factor` is reciprocal.

If its spin is not defined or zero, a neutron enters a magnetic field without

spin. In this case, the initial spin of the 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 enters the magnetic field with the defined spin direction irrespective of the direction of the magnetic field and polarization.

Table 5.29: Common source parameters (2)

parameter	explanation
`ispfs = 0, 1, 2`	(D=0) Neutron sources from spontaneous fission. If `ispfs=1 or 2` is specified in the `[source]` section and the following 18 nuclei are chosen as `proj`, neutrons can be defined using spontaneous fission as a source. These nuclei are assumed to be spontaneous fission nuclei: U-238, Pu-238, Pu-240, Pu-242, Cm-242, Cm-244, Cf-252, Th-232, U-232, U-233, U-234, U-235, U-236, Np-237, Pu-239, Pu-241, Am-241, Bk-249. `ispfs=1`: Tally result is normalized by the number of spontaneous fissions. `ispfs=2`: Tally result is normalized by the number of neutrons produced by spontaneous fission. In the case of `ispfs = 1,2`, `e0` and `e-type`, `dir`, and `a-type` are neglected. Unlike the case of RI source (`e-type=28,29`), particle type specified by `proj` is not used, namely, neutrons are generated as source particles.

When `ispfs=1,2` is specified, the multiplicity and energy spectrum of neutrons are taken from the reference [45]. The PHITS development team is grateful to Dr. Liem Peng Hong of NAIS, Co., Inc. for his support in developing this function to generate neutron sources.

In the next example, setting `proj`=240Pu and `ispfs`=1 selects neutrons from the spontaneous fission of Pu-240 as source particles. In this case, the value of `e0` is invalid.

```
[ Source ]
    s-type =   1
      proj =   240Pu
        e0 =   1.0
        z0 =   0
        z1 =   0
        r0 =   0.0
       dir =   1.0000
     ispfs = 1
```

[45] J. M. Verbeke, C. Hagmnn, and D. Wright, "Simulation of Neutron and Gamma Ray Emission from Fission and Photofission", UCRL-AR-228518 (2014).

5.3.3 Cylinder distribution source

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

Table 5.30: Parameters for cylinder source

`s-type = 1`	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 =`	(D=0.0) Minimum z of cylinder source[cm].
`z1 =`	(D=0.0) Maximum z of cylinder source[cm], (when z1=z0, circle plane source).
`r0 =`	(D=0.0) Radius of cylinder source, (when `r0=0.0`, a pencil source results)[cm].
`r1 =`	(D=0.0) Inner radius for inner void of cylinder.
`dir =`	(D=1.0) Direction cosine of projectile against z axis.
	If `all` is set, source is isotropic.
	If `data` is set, an `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 =`	(For mono-energy) Projectile energy[MeV/u].
`e-type =`	(For energy spectrum) Projectile energy[MeV/u].

5.3.4 Rectangular solid distribution source

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

Table 5.31: Parameters for rectangular solid source

`s-type = 2`	Rectangular solid and rectangle source.
`x0 =`	(D=0.0) Minimum x coordinate[cm].
`x1 =`	(D=0.0) Maximum x coordinate[cm].
`y0 =`	(D=0.0) Minimum y coordinate[cm].
`y1 =`	(D=0.0) Maximum y coordinate[cm].
`z0 =`	(D=0.0) Minimum z coordinate[cm].
`z1 =`	(D=0.0) Maximum z coordinate[cm], when (`z1=z0`), rectangle source.
`dir =`	(D=1.0) Direction cosine of projectile against z axis.
	If `all` is set, source is isotropic.
	If `data` is set, an `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 =`	(For mono-energy) Projectile energy[MeV/u].
`e-type =`	(For energy spectrum) Projectile energy[MeV/u].

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

The 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 omitted.

Table 5.32: Parameters for Gaussian source

`s-type = 3`	Gaussian source.
`x0 =`	(D=0.0) x coordinate of Gaussian center[cm].
`x1 =`	(D=0.0) FWHM in x direction[cm].
`y0 =`	(D=0.0) y coordinate of Gaussian center[cm].
`y1 =`	(D=0.0) FWHM in y direction[cm].
`z0 =`	(D=0.0) z coordinate of Gaussian center[cm].
`z1 =`	(D=0.0) FWHM in z direction[cm].
`dir =`	(D=1.0) Direction cosine of projectile against z axis. If `all` is set, source is isotropic. If `data` is set, an `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 =`	(For mono-energy) Projectile energy[MeV/u].
`e-type =`	(For energy spectrum) Projectile energy[MeV/u].

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

The generic parabola distribution consists 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 omitted.

Table 5.33: Parameters for generic parabola distribution

`s-type = 7`	Generic parabola source.
`x0 =`	(D=0.0) x coordinate of X-parabola center[cm].
`x1 =`	(D=0.0) X-parabola width[cm].
`y0 =`	(D=0.0) y coordinate of Y-parabola center[cm].
`y1 =`	(D=0.0) Y-parabola width[cm].
`z0 =`	(D=0.0) Minimum z of parabola[cm].
`z1 =`	(D=0.0) Maximum z of parabola[cm].
`rn =`	(D=2) Order of generic parabola.
`dir =`	(D=1.0) Direction cosine of projectile against z axis. If `all` is set, source is isotropic. If `data` is set, an `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 =`	(For mono-energy) Projectile energy[MeV/u].
`e-type =`	(For energy spectrum) Projectile energy[MeV/u].

5.3.7 Gaussian distribution source (x-y plane)

The source of the Gaussian distribution in x-y plane can be set. 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 omitted.

Table 5.34: Parameters for Gaussian source

`s-type = 13`	Gaussian source.
`x0 =`	(D=0.0) *x* coordinate of Gaussian center[cm].
`y0 =`	(D=0.0) *y* coordinate of Gaussian center[cm].
`r1 =`	(D=0.0) FWHM in *xy* plane[cm].
`z0 =`	(D=0.0) Minimum *z* coordinate[cm].
`z1 =`	(D=z0) Maximum *z* coordinate[cm].
`dir =`	(D=1.0) Direction cosine of projectile against *z* axis. If `all` is set, source is isotropic. If `data` is set, an `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 =`	(For mono-energy) Projectile energy[MeV/u].
`e-type =`	(For energy spectrum) Projectile energy[MeV/u].

5.3.8 Generic parabola distribution source (x-y plane)

The source of the generic parabola distribution in x-y plane can be set. 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 omitted.

Table 5.35: Parameters for generic parabola distribution

`s-type = 15`	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 =`	(D=0.0) Parabola width in *xy* plane[cm].
`z0 =`	(D=0.0) Minimum *z* of parabola[cm].
`z1 =`	(D=0.0) Maximum *z* of parabola[cm].
`rn =`	(D=2) order of generic parabola.
`dir =`	(D=1.0) Direction cosine of projectile against *z* axis. If `all` is set, source is isotropic. If `data` is set, an `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 =`	(For mono-energy) Projectile energy[MeV/u].
`e-type =`	(For energy spectrum) Projectile energy[MeV/u].

5.3.9 Sphere and spherical shell distribution source

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

Table 5.36: Parameters for sphere and spherical shell source

`s-type = 9`	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 =`	(D=0.0) Inside radius[cm]. If `r1=0`, sphere source.
`r2 =`	(D=0.0) Outside radius[cm].
`dir =`	(D=1.0) 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 orthogonal to a circle of radius `r2` toward the sphere.
`e0 =`	(For mono-energy) Projectile energy[MeV/u].
`e-type =`	(For energy spectrum) Projectile energy[MeV/u].

When using the source type `s-type=9` for volume and area calculation, set `dir=-all, r1=r2`; `dir=iso` gives the same result.

Inner radius, `r1`, should be smaller or equal to outer radius, `r2`, except for `dir=iso`, where `r1` should be equal to (or larger than) `r2`.

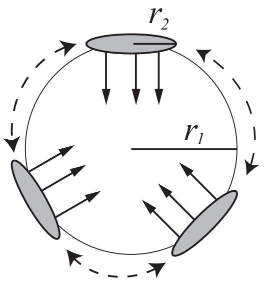


Figure 5.3: Schematic of the source in the case of `dir=iso`.

5.3.10 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 omitted.

Table 5.37: 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 =`	(D=0.0) Ratio of (maximum radius)/(minimum radius) for *x* direction[cm/mrad].
`y0 =`	(D=0.0) *y* coordinate of beam center[cm].
`y1 =`	(D=0.0) Ratio of (maximum radius)/(minimum radius) for *y* direction[cm/mrad].
`z0 =`	(D=0.0) Minimum *z*[cm].
`z1 =`	(D=0.0) Maximum *z*[cm].
`rx =`	(D=0.0) Gradient of ellipse in a phase space on *x* direction[rad].
`ry =`	(D=0.0) Gradient of ellipse in a phase space on *y* direction[rad].
`wem =`	(D=0.0) Emittance (π cm × mrad).
`dir =`	(D=1) Direct cosine. (1 or -1 only).
`e0 =`	(For mono-energy) Projectile energy[MeV/u].

5.3.11 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 omitted.

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 5.38: 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 =`	(D=1) Direct 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 5.39: 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).

5.3.12 Cone shape

Parameters for cone shape source are shown below. In this case, source particles are generated on the side surface of the cone. The order of parameters is free. If a parameter has a default value (D=***), the parameter can be omitted.

Table 5.40: Parameters for cone shape

s-type = 18	Cone shape.
x0 =	(D=0.0) x coordinate offset of top of cone[cm].
y0 =	(D=0.0) y coordinate offset of top of cone[cm].
z0 =	(D=0.0) z coordinate offset of top of cone[cm].
x1 =	(D=0.0) x-component of vector from top to center of bottom circle[cm].
y1 =	(D=0.0) y-component of vector from top to center of bottom circle[cm].
z1 =	(D=0.0) z-component of vector from top to center of bottom circle[cm].
r0 =	(D=0.0) distance between top and upper end of source locations on axis of cone[cm].
r1 =	(D=0.0) Distance between top and lower end of source locations on axis of cone[cm].
r2 =	(D=0.0) Angle between generatrix and axis of cone[degree].
dir =	(D=1.0) Direction cosine of projectile against z axis. If `all` is set, source is isotropic. If `data` is set, an `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 =	(For mono-energy) Projectile energy[MeV/u].
e-type =	(For energy spectrum) Projectile energy[MeV/u].

5.3.13 Triangle prism shape

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

Table 5.41: Parameters for triangle prism shape

`s-type = 20`	Triangle prism shape.
`x0 =`	(D=0.0) x-coordinate of origin apex of triangle[cm].
`y0 =`	(D=0.0) y-coordinate of origin apex of triangle[cm].
`z0 =`	(D=0.0) z-coordinate of origin apex of triangle[cm].
`x1 =`	(D=0.0) x vector from the origin to the first apex[cm].
`y1 =`	(D=0.0) y vector from the origin to the first apex[cm].
`z1 =`	(D=0.0) z vector from the origin to the first apex[cm].
`x2 =`	(D=0.0) x vector from the origin to the second apex[cm].
`y2 =`	(D=0.0) y vector from the origin to the second apex[cm].
`z2 =`	(D=0.0) z vector from the origin to the second apex[cm].
`x3 =`	(D=0.0) x vector from the origin to the third apex[cm].
`y3 =`	(D=0.0) y vector from the origin to the third apex[cm].
`z3 =`	(D=0.0) z vector from the origin to the third apex[cm].
`exa =`	(D=0.0) Attenuation coefficient. exa>0; $\exp(-ax)$, exa=0; uniform.
`dir =`	(D=1.0) Direction cosine of projectile against z axis. If `all` is set, source is isotropic. If `data` is set, an `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 =`	(For mono-energy) Projectile energy[MeV/u].
`e-type =`	(For energy spectrum) Projectile energy[MeV/u].

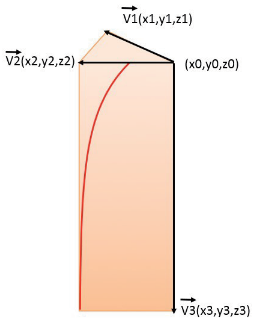


Figure 5.4: Coordinates of triangle prism

5.3.14 xyz-mesh distribution source

By specifying `s-type=22`, a complex spatial distribution of sources can be set. In `stype=22`, mono-energetic particles can be generated. The spatial distribution can be defined by `x,y,z-type` subsections and data of relative intensities $I_{ijk}, (i = 1, \cdots, n_x; j = 1, \cdots, n_y; k = 1, \cdots, n_z)$. The data should be written below the subsections. The direction of the sources can be specified by `dir, phi`, and `dom`. Note that the direction in each xyz-bin cannot be changed. Parameters for xyz-mesh distribution source are shown in Table 5.42. The order of parameters is free except for the data of the relative intensities. If a parameter has a default value (D=`***`), the parameter can be omitted.

Table 5.42: Parameters for xyz-mesh distribution source

`s-type = 22`	*xyz*-mesh distribution source.
`mesh = xyz`	Geometry mesh. Only `xyz` can be set.
`x-type =`	Mesh type for *x*-axis. `x-type` subsection is needed below this option.
`y-type =`	Mesh type for *y*-axis. `y-type` subsection is needed below this option.
`z-type =`	Mesh type for *z*-axis. `z-type` subsection is needed below this option.
	Put data of relative intensities I_{ijk} below `x, y, z-type` subsections. The change of the data element should be in the order of x, y, z. Note that the change in x, y, z axes should be ascending order if `nx, ny, nz` are positive, and it should be descending order if `nx, ny, nz` are negative.
`dir =`	(D=1.0) Direction cosine of projectile against *z* axis. If `all` is set, source is isotropic. If `data` is set, an `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 =`	(For mono-energy) Projectile energy[MeV/u].
`e-type =`	(For energy spectrum) Projectile energy[MeV/u].

When all of `nx,ny,nz` are positive, the order of the relative intensities I_{ijk} is given as

$$I_{111}, I_{211}, \cdots, I_{n_x 11}, I_{121}, I_{221}, \cdots, I_{n_x n_y 1}, I_{112}, I_{212}, \cdots, I_{n_x n_y n_z}, \tag{2}$$

where $n_x =$`|nx|`, $n_y =$`|ny|`, $n_z =$`|nz|`. When `nx` and `ny` are positive and `nz` is negative, the order is given as

$$I_{11n_z}, I_{21n_z}, \cdots, I_{n_x 1 n_z}, I_{12n_z}, I_{22n_z}, \cdots, I_{n_x n_y n_z}, I_{11n_z-1}, I_{21n_z-1}, \cdots, I_{n_x n_y 1}. \tag{3}$$

In the latter case, the data with $k = n_z =$`|nz|` should be first put.

An example using `s-type=22` is shown below.

Example 2: Example of xyz-mesh distribution source

```
[ Source ]
 s-type = 22
   proj = neutron
     e0 = 1.0
    dir = all
   mesh = xyz
 x-type = 2
     nx = 3
   xmin = -10
   xmax =  10
 y-type = 2
     ny = -3
   ymin = -10
   ymax =  10
 z-type = 2
     nz = -2
   zmin = -10
```

```
18:      zmax =   10
19:        1 2 3
20:        4 5 6
21:        7 8 9
22:
23:        1 0 3
24:        0 5 0
25:        7 0 9
```

In this case, the source region is [-10cm≤ x, y, z ≤10cm], and the ranges in x, y, and z-axes are divided by 3, 3, and 2 bins, respectively. Because `ny` and `nz` are negative, the relative intensities I_{ijk} should be written in descending order for j and k. Therefore, in the 19th-21st lines of the example, I_{ijk} with $k = 2$, i.e. in the region of [0cm≤ z ≤10cm], are given as

$$\begin{pmatrix} I_{132} & I_{232} & I_{332} \\ I_{122} & I_{222} & I_{322} \\ I_{112} & I_{212} & I_{312} \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}, \qquad (4)$$

and I_{ijk} with $k = 1$, i.e. in the region of [-10cm≤ z ≤0cm], are given in the 23rd -25th lines as

$$\begin{pmatrix} I_{131} & I_{231} & I_{331} \\ I_{121} & I_{221} & I_{321} \\ I_{111} & I_{211} & I_{311} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 3 \\ 0 & 5 & 0 \\ 7 & 0 & 9 \end{pmatrix}. \qquad (5)$$

Figure 5.5 shows the sources distribution of the example, which are obtained by `[t-product]` with `output=source`. The left and right panels are results in the regions of [0cm≤ z ≤10cm] and [-10cm≤ z ≤0cm], respectively. In the left panel, the regions of 1, 2, 3, 4, 5, 6, $\cdots$ correspond to $I_{132}, I_{232}, I_{332}, I_{122}, I_{222}, I_{322}, \cdots$, respectively. As the region number increases, the intensity in each region gradually increases. In the right panel, the sources don't generate in the regions of 2, 4, 6, and 8, which correspond to the elements of 0 in the right side of Eq. 5.

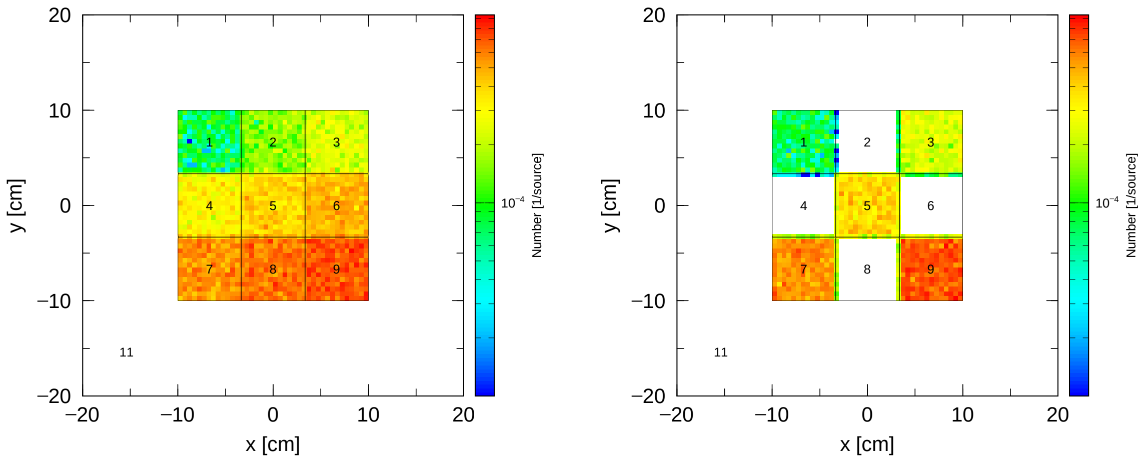


Figure 5.5: Results of example 2 in the region of [0cm≤ z ≤10cm] (left panel) and [-10cm≤ z ≤0cm] (right panel).

5.3.15 Tetra-mesh source

By specifying `s-type=24`, particle generation from tetrahedrons defined in tetrahedron geometry can be considered (sec. 5.6.5). Specification of the cell number of the universe where tetrahedrons belong is needed. Particles are produced uniformly from each tetrahedron which belongs to the specified universe with a rate proportional to its volume. Here only one cell number can be specified and therefore multi-source (sec. 5.3.1) have to be used to select more than one cell. The direction of the sources can be specified by `dir, phi`, and `dom`. Parameters for tetra-mesh source are shown in Table 5.43. The order of parameters is free. If a parameter has a default value (D=***), the parameter can be omitted. Please refer also ppt files and sample input files in `\utility\TetraGEOM`.

Table 5.43: Parameters for tetra-mesh source

`s-type = 24`	Tetra-mesh source.
`tetreg =`	The cell number of the universe where tetrahedrons belong. Only one cell number can be specified.
`dir =`	(D=1.0) Direction cosine of projectile against *z* axis. If `all` is set, source is isotropic. If `data` is set, an `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 =`	(For mono-energy) Projectile energy[MeV/u].
`e-type =`	(For energy spectrum) Projectile energy[MeV/u].

An example using `s-type=24` is shown below.

Example 3: Example of tetra-mesh source

```
[ Source ]
 s-type = 24
   proj = proton
 tetreg = 201
    dir = all
     e0 = 70
```

In this case, source particles are produced uniformly from the region of the cell number 201 specified by `tetreg`. This specified cell number should belong the universe as the 13th line of the example 44.

5.3.16 Surface definition source

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

Table 5.44: Parameters for surface definition source

`s-type = 26`	Surface source.
`suf =`	Surface number for generating sources defined in `[surface]` section. Only one surface can be defined for this parameter. The types of the surfaces that can be used for this parameter are `PX, PY, PZ, SO, S, SX, SY, SZ, C/X, C/Y, C/Z, CX, CY, CZ`.
`cut =`	Surface numbers to restrict the source generation area on the surface. The format is the same as `[cell]` section, but operators such as `( )` and `:` cannot be used. For example, if you define `cut = 1 -2 3 -4`, the source area is restricted to positive side (or outside) of surface 1 & 3, and negative side (or inside) of surface 2 & 4.
`dir =`	(D=1.0) Direction cosine of projectile against the positive normal line of the source surface. If `all` is set, source is isotropic. If `data` is set, an `a-type` subsection is necessary.
`phi =`	(D=all) Azimuthal angle[degree].
`dom =`	(D=0.0) Solid angle[degree]. $= -1$; $\cos^2$ bias distribution.
`e0 =`	(For mono-energy) Projectile energy[MeV/u].
`e-type =`	(For energy spectrum) Projectile energy[MeV/u].

5.3.17 Reading dump file

By setting `s-type=17`, information on particles recorded in a file can be used as sources. The "dump" file can be obtained by using the dump option in the `[t-cross]`, `[t-product]`, and `[t-time]` tallies. There are two alternative methods for treating the dump data, which can be switched between using the `idmpmode` parameter: by setting `idmpmode=0`, each particle's data recorded in the dump file can treated as an independent history; by setting `idmpmode=1`, the correlations between particles at the time of data dump can be taken into account via statistical processing based on history information recorded in the dump file. For continuous calculation using the dump file based on the history information from the previous step, the use of `idmpmode=1` is recommended. [46]

Parameters for `s-type=17` are shown below. The order of parameters is free. If a parameter has a default value (D=`***`), the parameter can be omitted.

Table 5.45: Parameters for dump file source (1)

`s-type = 17`	Reading dump file.
`file =`	Dump filename (with full pathname).
`dump =` (next line)	Dump data number; if this is negative, data are written as Ascii. identification of dump data.
(omitted)	If the following parameters are specified, these values have priority over the dump data. If the dump data do not include the following data, the relevant parameters should be specified.
`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 of projectile against *z* axis. If `all` is set, source is isotropic. If `data` is set, an `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 =`	(For mono-energy) Projectile energy[MeV/u].
`e-type =`	(For energy spectrum) Projectile energy[MeV/u].
`wgt =`	(D=1.0) Weight of source particle.

[46] The option `idmpmode=1` and re-used calculation using the option `dmpmulti` were introduced by referring to the presentation "Estimation of uncertainty in multi-step Monte Carlo calculation" (N50) by Dr. Y. Namito *et al.* at the 2015 annual meeting of AESJ (Hitachi, Japan).

Table 5.46: Parameters for dump file source (2)

(omitted)	The following parameters can also be specified.
factor =	(D=1.0) normalization of source particle.
t-type = reg = ntmax = trcl =	(D=0) time distribution. (D=all) specify the region. (D=1000) maximum re-try number when reg is specified. (D=none) transform number, or definition of transform.
idmpmode =	Control parameter to select dump source mode. (D=0) (nocas and nobach) not included in the case history information dump data. (D=1) (nocas and nobach) included in the case history information dump data. =0: Each particle data recorded in the dump file is treated as an independent history. =1: Particle data in the dump file are processed by taking account of particle correlations in the previous step.
dmpmulti =	Control parameter for reuse of dump file. (D=0.0) With idmpmode=0. (D=1.0) With idmpmode=1. =0.0: The dump file is rewound and re-used until total histories are finished, as specified by (maxcas*maxbch). However, this mode cannot be used with idmpmode=1. >0.0: The dump file is re-used the number of times specified by this parameter. The digits after the decimal point reflect the probability with which the Russian roulette routine will adopt another reproduction of the particle data.

To use idmpmode=1, the history number (nocas) and batch number (nobch) must be contained in the dump data. Using the PHITS tally dump option, a dump file named "***_dmp" is created in addition to the normal tally file with the file name "***" specified by file=***. Both files are required to use idmpmode=1 and must be placed in the same folder. The total number of histories of the previous step is adopted from the values of maxcas and maxbch written in this tally file and the history numbers, maxcas and maxbch, given in the PHITS input file are neglected. The normalization factor totfact is ignored when idmpmode=1. To use the normalization factor, totfact should be applied to the previous step for creating the dump file, at which point it will be reflected in the weights of particles recorded in the dump file. The option idmpmode=1 is incompatible with multi-source processing.

The option dmpmulti controls the number of re-uses of the dump file. For instance, in the case of dmpmulti=2.0, the dump file will be reused twice, i.e., information for each particle recorded in the dump file is reproduced twice and the particle is treated as two particles having half-weights based on a different random number. The digits after the decimal point indicate the probability of creating another reproduction of the particle as stochastically determined by running the Russian roulette procedure for each particle. For example, in the case of dmpmulti=2.3, the particle data are reproduced twice with probability 70 % and three times with probability 30 %. As a special mode, setting dmpmulti=0.0 rewinds and re-uses the dump file until the total histories specified by (maxcas*maxbch) in the PHITS input file are finished. This mode is valid only with idmpmode=0.

The restart calculation (istdev<0) is essentially not applicable for dump source calculation; it only allowed with dmpmulti=0.0, in which case it should be noted that the calculation might give biased results unless the dump file contains sufficiently large particle data. Thus, re-calculation using a larger dmpmulti factor is recommended to achieve better statistics; if the particle data contained in the dump file are not statistically suffcient, the statistical uncertainties may not satisfactorily diminish even if a large dmpmulti factor is selected. In this case, it will be necessary to re-calculate from the previous step to create suffcient dump data.

The parameter dump specifies the number of dump data points per record. If this number is positive, the data are read as binary data; setting dump as a negative value will cause the data to be read as ascii data. In the line following the dump= assignment, the data sequence of the record is input. The respective dump file physical quantities are encoded with the id numbers shown in Tables 5.47 and 5.48.

Table 5.47: Dump data id number (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 5.48: Dump data id number (2)

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

In the table, kf is the particle kf-code (see Table 4.4), x, y, and z are coordinates [cm], u, v, and w are the unit vectors of particle coordinate system, e is the energy ([MeV], or [MeV/u] for nuclei), wt is the weight, time is the initial time [ns], c1, c2, and c3 are the counter values, and sx, sy, and sz are the unit vectors of the respective spin directions. In Table 4.43, which encodes history-related ids, name is the particle's collision number, nocas is the batch history number, nobch is the batch number, and no is the cascade id contained in the history. These are assumed to in real*8 format for binary data and in *n*(1p1e24.15) data format for ascii data.

For an example, a record containing nine data points is given by

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

To read these data, the parameters are written as

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

5.3.18 User definition source

An original source function can be used by editing “usrsors.f,” and setting `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 omitted.

Table 5.49: 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 of projectile against z axis. If `all` is set, source is isotropic. If `data` is set, an `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 =`	(For mono-energy) Projectile energy[MeV/u].
`e-type =`	(For energy spectrum) Projectile energy[MeV/u].
`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.

User-defined variables `c1-c99` by setting `set` can be used in “usrsors.f” as `cval(1-99)`. Note that the last setting of `set` is available if the same variable is defined several times.

A sample program of “usrsors.f” is shown 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 a 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.

File 2: 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 )
      data ifirst / 0 /
      save ifirst
      character filenm*50
*-----------------------------------------------------------------------
*        example of initialization
*-----------------------------------------------------------------------
         if( ifirst .eq. 0 ) then
c              filenm = 'input.dat'
c              inquire( file = filenm, exist = exex )
c              if( exex .eqv. .false. ) then
c                 write(*,*) 'file does not exist =>  ', filenm
c                 call parastop( 887 )
c                 end if
c              open(71, file = file(i), status = 'old' )

c              close(71)
               ifirst = 1
         end if
*-----------------------------------------------------------------------
*        example for 3 GeV proton with z-direction
*-----------------------------------------------------------------------
            x = 0.0
            y = 0.0
            z = 0.0
            u = 0.0
            v = 0.0
            w = 1.0
            e = 3000.0
           wt = 1.0
         time = 0.0
         name = 1
           kf = 2212
          nc1 = 0
          nc2 = 0
          nc3 = 0
           sx = 0.d0
           sy = 0.d0
           sz = 0.d0
*-----------------------------------------------------------------------
      return
      end
```

5.3.19 Definitions for energy distribution

The energy-distributed source can be defined by `e-type` subsection instead of by `e0`. Note that `e0` have to be deleted or set to invalid by comment marks when the `e-type` subsection is given. Attention should also be paid to the unit of different kinds of source particles. Throughout this section, the unit is [/(MeV/u)] for ion sources (d, t, α, and nuclei), and [/MeV] for other types of sources.

Available types of the source energy distributions are shown in Table 5.50. There are integral and differential types as the format of the distribution; for the integral type, the source intensity should be given as an energy-integrated value, while for the differential type, the intensity should be given as the energy differential value expressed in [particles/MeV] (particles/MeV/u for ions). As generation method of the source particles, two methods can be selected. In the first one, the number of particles generated in a bin is proportional to the intensity, and the specified energy distribution is statistically described. In the second one, the number of source particles generated in each bin is the same for all energy bins, but the weight values of source particles are adjusted to be proportional to the intensity.

Table 5.50: Types of source energy distribution. There are two methods for generation of source particles; one is to adjust statistically the number of particles in each energy bin with the constant weight value, while the other is to change the weight value of the source particle in each bin with the constant number of particles. When setting the numbers in the brackets to `e-type`, the energy can be given by wave length (Å).

Type	Generation method	`e-type`	Explanation
Continuous energy distribution with integral type	Number of particles Weight value	1, (11) 4, (14)	A continuous energy distribution can be specified by providing the data set of lower limits of energy bins and integrated values of the particle generation probabilities in each bin.
Continuous energy distribution with differential type	Number of particles Weight value	21, (31) 24, (34)	A continuous energy distribution can be specified by providing the data set of the minimum of energy bins and differential probabilities of the particle generation in each bin.
Discrete energy distribution	Number of particles Weight value	8, (18) 9, (19)	A discrete energy distribution can be specified by giving data set of energy points and probabilities of the particle generation at each points.
Discrete+Continuous energy distribution with integral type	Number of particles Weight value	22, (32) 23, (33)	Any energy distribution can be specified by providing the data set of the minimum and maximum of energy bins, and integrated values of particle probabilities in each bin.
Gaussian distribution with differential type	Number of particles	2, (12)	A differential spectrum of Gaussian distribution can be specified by giving a mean value and FWHM of Gaussian.
Maxwellian distribution with differential type	Number of particles Weight value	3 7	A differential spectrum of Maxwellian distribution can be specified by giving a temperature parameter of Maxwellian.
Analytical function with differential type	Number of particles Weight value	5, (15) 6, (16)	A differential spectrum can be specified by giving analytical functions.
Energy distribution for RI source	Number of particles Weight value	28 29	The α, β, and γ-rays of radioisotope (RI) decay are generated by simply specifying the activity and names of the RIs.
Energy distribution of tally results	Number of particles	20	A energy distribution can be specified by using results of tallies.

Input formats and parameters in each `e-type` will be explained below. If a parameter has a default value (D=***), the parameter can be omitted.

Continuous energy distribution with integral value

Table 5.51: Parameters for source energy distribution (1)

parameter	explanation
`e-type = 1, (11)`	Any energy distribution can be specified by providing the data set of energy bins `e(i)` and the integrated values of the particle generation probabilities `w(i)` by hand. The number of particles generated in a bin is proportional to `w(i)`, and the specified energy distribution is statistically described. For case 11, the energy is given by the wave length (Å).
`ne =`	Number of energy groups. If this is a positive number, source particles are generated so that the energy differential fluxes in units of [1/MeV] become constant in each bin. If negative, the fluxes in units of [1/Lethargy] become constant for each bin. Data must be provided in the following line using the format `(e(i), w(i), i=1, ne), e(ne+1)`. The integrated numbers of particles generated in each energy bin is proportional to `w(i)`.
`e-type = 4, (14)`	The same energy distribution as in the case of `e-type = 1, (11)` can be specified, except with the energy bins `e(i)` and weights of the source particle `w(i)` entered by hand. The number of source particles generated in each bin is the same for all energy bins, but integrated values of the weights of source particles are adjusted to be proportional to `w(i)`. The number of source particles generated in each bin can also be changed by specifying `p(i)`. For case 14, the energy is given by wave length (Å).
`ne =`	Number of energy groups. If this is a positive number, source particles are generated so that the energy differential fluxes in units of [1/MeV] become constant in each bin. If negative, the fluxes in units of [1/Lethargy] become constant for each bin. Data must be provided in the following line using the format `(e(i), w(i), i=1, ne), e(ne+1)`. In the default (`p-type = 0`), equal numbers of particles are generated in each cell. The integrated number of source particles generated in each 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 using the format `(p(i),i=1,ne)`.

An alternative option for neutron optics has been developed to specify the energy as a wavelength: if `e-type` is set to `= 11, 12, 14`, the wavelength (Å) is used as the energy unit. In the other cases, the expression e0 = 8.180425e-8/13**2 is used, which corresponds to the energy of a neutron with a 13Å wavelength.

In the case of `e-type=1`, the format is as follows.

```
e-type = 1
    ne = n
       e(1)   w(1)
       e(2)   w(2)
       e(3)   w(3)
       ....   ...
       e(n-1) w(n-1)
       e(n)   w(n)
       e(n+1)
```

In this case, the energy distribution is given as

```
e(1)-e(2)   w(1)
e(2)-e(3)   w(2)
e(3)-e(4)   w(3)
.......     ...
e(n-1)-e(n) w(n-1)
e(n)-e(n+1) w(n)
```

As an example, use the following code to set the strengths of three energy bins between 0-2 MeV, 2-4 MeV, and 4-6 MeV as 0.2, 0.6, and 0.2, respectively:

```
e-type = 1
    ne = 3
       0   0.2
       2   0.6
       4   0.2
       6
```

Continuous energy distribution with differential value

Table 5.52: Parameters for source energy distribution (2)

parameter	explanation
e-type = 21, (31)	Any energy distribution can be specified by giving data set of energy bins e(i) and differential probabilities of the particle generation dφ/dE(i) by hand. The integrated number of the particle generation in the bin is proportional to dφ/dE(i)*{e(i+1)-e(i)}, and the specified energy distribution is statistically described. For 31 case, energy is given by wave length (Å).
ne =	Number of energy group. If it is given by positive number, source particles are generated so that the energy differential fluxes in the unit of [1/MeV] become constant in each bin. On the other hand, if ne is negative, the fluxes in the unit of [1/Lethargy] become constant in each bin. Data must be given from the next line by the format as (e(i),dφ/dE(i),i=1,ne), e(ne+1). The integrated number of the particle generation in the each energy bin is proportional to dφ/dE(i)*{e(i+1)-e(i)}.
e-type = 24, (34)	The same energy distribution as in the case of e-type=21,(31) can be specified. Unlike e-type=21,(31), the distribution is described by giving data set of energy bins e(i) and weights of the source particle w(i) by hand. The number of source particles generated in each bin is the same for all energy bin, but integrated values of the weight of source particles are adjusted to be proportional to w(i)*{e(i+1)-e(i)}. The number of source particles generated in each bin can also be changed by specifying p(i). For 34 case, energy is given by wave length (Å).
ne =	Number of energy group. If it is given by positive number, source particles are generated so that the energy differential fluxes in the unit of [1/MeV] become constant in each bin. On the other hand, if ne is negative, the fluxes in the unit of [1/Lethargy] become constant in each bin. Data must be given from the next line by the format as (e(i),w(i),i=1,ne), e(ne+1). In default (p-type=0), equal number of particle is generated in each cell. The integrated number of source particles generated in each 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).

Discrete energy distribution

Table 5.53: Parameters for source energy distribution (3)

parameter	explanation
`e-type = 8, (18)`	Any energy distribution can be specified by giving data set of energy points `e(i)` and probabilities of the particle generation `w(i)` by hand. The number of the particle generation at the point is proportional to `w(i)`, and the specified energy distribution is statistically described. For 18 case, energy is given by wave length (Å).
`ne =`	Number of energy points. Data must be given from the next line by the format as `(e(i),w(i),i=1,ne)`. The number of the particle generation at the each energy point is proportional to `w(i)`.
`e-type = 9, (19)`	The same energy distribution as in the case of `e-type=8,(18)` can be specified. Unlike `e-type=8,(18)`, the distribution is described by giving data set of energy points `e(i)` and weights of the source particle `w(i)` by hand. The number of source particles generated in each point is the same for all energy point, but integrated values of the weight of source particles are adjusted to be proportional to `w(i)`. The number of source particles generated in each bin can also be changed by specifying `p(i)`. For 19 case, energy is given by wave length (Å).
`ne =`	Number of energy points. Data must be given from the next line by the format below. In default (`p-type=0`), equal number of particle is generated at each point. `(e(i),w(i),i=1,ne)` The number of source particles generated in each point 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)`.

Energy distribution by free format

Table 5.54: Parameters for source energy distribution (4)

parameter	explanation
`e-type = 22, (32)`	Any energy distribution can be specified by providing the data set of the minimum `emin(i)` and maximum `emax(i)` of energy bins, and the integrated values of the particle probabilities `w(i)` by hand. To define the same discrete distribution as `e-type=18, (28)`, set the maximum to the same value as the minimum. The number of particles generated in a bin is proportional to `w(i)`, and the specified energy distribution is statistically described. For case 32, the energy is given by the wave length (Å).
`ne =`	Number of energy groups. If this is a positive number, source particles are generated so that the energy differential fluxes in units of [1/MeV] become constant in each bin. If negative, the fluxes in units of [1/Lethargy] become constant for each bin. Data must be provided in the following line using the format: `(emin(i),emax(i),w(i),i=1,|ne|)`. The integrated numbers of particles generated in each energy bin is proportional to `w(i)`.
`e-type = 23, (33)`	The same energy distribution as in the case of `e-type=22,(32)` can be specified, except with the weights of the source particle `w(i)` entered by hand. The number of source particles generated in each bin is the same for all energy bins, but integrated values of the weights of source particles are adjusted to be proportional to `w(i)`. The number of source particles generated in each bin can also be changed by specifying `p(i)`. For case 33, the energy is given by wave length (Å).
`ne =`	Number of energy groups. If this is a positive number, source particles are generated so that the energy differential fluxes in units of [1/MeV] become constant in each bin. If negative, the fluxes in units of [1/Lethargy] become constant for each bin. Data must be provided in the following line using the format: `(emin(i),emax(i),w(i),i=1,|ne|)`. In the default (`p-type=0`), equal numbers of particles are generated in each cell. The integrated number of source particles generated in each 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)`.

An alternative option for neutron optics has been developed to specify the energy as a wavelength: if `e-type` is set to = `32, 33`, the wavelength (Å) is used as the energy unit. In the other cases, the expression e0 = 8.180425e-8/13**2 is used, which corresponds to the energy of a neutron with a 13Åwavelength.

In the case of `e-type = 22`, the format is as follows:

```
e-type = 22
    ne = n
       emin(1) emax(1)   w(1)
       emin(2) emax(2)   w(2)
       emin(3) emax(3)   w(3)
       ....   ...
       emin(n-1) emax(n-1) w(n-1)
       emin(n) emax(n)   w(n)
```

As an example, use the following code to set the strengths of three energy bins between 0-2 MeV, 2-4 MeV, and 4-6 MeV as 0.2, 0.6, and 0.2, respectively, and the discrete one of mono-energy of 5.6 MeV as 0.4:

```
e-type = 22
    ne = 4
       0   2   0.2
       2   4   0.6
       4   6   0.2
       5.6 5.6 0.4
```

Gaussian energy distribution

Table 5.55: Parameters for source energy distribution (5)

parameter	explanation
`e-type = 2, (12)`	Differential spectrum `d`φ`/dE(i)` is given by 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`	Differential spectrum `d`φ`/dE(i)` is given by Maxwellian distribution: $f(x) = x\exp(-x/T)$
`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.

`et0 =`
`et1 =`
`et2 =`
	temperature parameter T [MeV].
	minimum cut off for Maxwellian distribution [MeV].
	maximum cut off for Maxwellian distribution [MeV].

`e-type = 7`	The same energy distribution as in the case of `e-type=3` can be specified. Unlike `e-type=3`, the number of source particles generated in each bin is the same for all energy bin, but integrated values of the weight of source particles are adjusted to be proportional to $f(x) = x^{1.5}\exp(-x/T)$. The number of source particles generated in each bin can also be changed by specifying `p(i)`.
`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 the default (`p-type=0`), equal numbers of particles are generated in each cell. The integrated number of source particles generated in each bin is proportional to `p(i)`.

`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,nm)`.

Energy distribution defined by analytical functions

Table 5.56: Parameters for source energy distribution (6)

parameter	explanation
`e-type = 5, (15)`	Differential spectrum `d`φ`/dE(i)` is given by $f(x)$. For 15 case, energy is given by wave length (Å).
`f(x) =`	Any analytical function of x, Fortran style. x denotes energy [MeV/u]. One can use intrinsic functions and constants `C`, e.g., `f(x) = exp(-c1*x**2)`.
`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. Integrated number of source particles generated in each 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)`	The same energy distribution as in the case of `e-type=5,(15)` can be specified. Unlike `e-type=5,(15)`, the number of source particle generated in each bin is the same for all energy bin, but integrated values of the weight of source particles are adjusted to be proportional to $f(x)$. The number of source particles generated in each bin can also be changed by specifying `p(i)`. For 16 case, energy is given by wave length (Å).
`f(x) =`	Any analytical function of x, Fortran style. x denotes energy (MeV/u). One can use intrinsic functions and constants `C`, e.g., `f(x) = exp(-c1*x**2)`.
`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 the default (`p-type=0`), equal numbers of particles are generated in each cell. The integrated number of source particles generated in each bin is proportional to `p(i)`.

`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)`.

Energy distribution of RI source

Table 5.57: Parameters for source energy distribution (7)

parameter	explanation
`e-type= 28,29`	The α, β (including Auger electrons), and γ-rays of radioisotope (RI) decay are generated by simply specifying the activity (in Bq) and names of the RIs. The DECDC[47]nuclear decay database (equivalent to ICRP107) is used to obtain the energy spectra. To use this function, the directory containing the DECDC data file "RIsource.datd" must be specified by setting `file(24) (D=c:\phits\data\)` in the `[parameters]` section. When `e-type=28`, spectra are expressed by changing the probabilities of generating particle energy spectra obtained from DECDC. When `e-type=29`, spectra are expressed by changing the weights of the source particles while maintaining constant probabilities for all source energy.
`ni=`	Number of RIs. RI names and activity must be given in the next line using the format `(RI(i),A(i),i=1, ni)`. `RI(i)` can be defined using one of two formats, e.g., as 137Cs or Cs-137. `A(i)` is specified in units of Bq.
`dtime=`	(D=−10.0) Option for time evolution [s]. `dtime>0` (in s): The energy spectrum is determined based on the decays of specified RIs including contributions from their daughter nuclides at the time at which `dtime` has passed; this changes the activity of each RI from its specified value. For example, setting 100 Bq for an RI with a half-life of 1 min and `dtime=60` results in an RI activity of 50 Bq in the PHITS calculation. `dtime=0`: No time evolution is considered. `dtime<0`: The energy spectrum is determined based on the decays of the specified RIs including the contributions from their daughter nuclides at the time at which the half-life×|`dtime`| has passed. Unlike the case of `dtime>0`, the activities of the respective RI are unchanged from their specified values. For example, setting 100 Bq for an RI with a half-life of 1 min and `dtime`=−1.0 results in an unchanged RI activity of 100 Bq in the PHITS calculation. When no information on the time necessary to attain radioactive equilibrium is present, a large negative value, e.g., `dtime`=−10.0, must be set. It is not possible to specify both parent and daughter nuclides simultaneously for `dtime<0`.

To use this function with multi-sources, set <`source`>=`1.0` and `totfact` as follows:

- when all sources are defined by `e-type=28,29`, set totfact as the number of <`source`>;
- when the multi-sources include sources defined by `e-type≠28,29`, set each <`source`> as the absolute intensity of each RI and set `totfact` as the sum of all values of <`source`>.

If the `totfact` is given as a negative value, the same particle is generated in each multi-source section with weights adjusted according to the ratio of the activities.

[47] A. Endo, Y. Yamaguchi and K.F. Eckerman, Nuclear decay data for dosimetry calculation - Revised data of ICRP Publication 38, JAERI 1347 (2005).

Table 5.58: Parameters for source energy distribution (8)

parameter	explanation
	(continued in the cases of e-type= 28,29)
actlow=	(D=1.0e-10) Option for lower limit of the activity [Bq]. If the activity following the passage of the specified time is smaller than actlow, the decay of the corresponding RI is not included in the sources.
norm=	(D=0) Option for normalization of tally results. 0: Tally results are normalized in units of [/s]. In this case, the source intensity is determined by the specified activity (if dtime>0, its decrease owing to decay is also taken into account). 1: Tally results are normalized in units of [/source]. This is equivalent to the general normalization of PHITS.
iaugers=	(D=0) Option for the production of Auger & internal conversion (IC) electrons. (only for proj=electron) 0: All electrons (β-rays, Auger & IC electrons) are generated. 1: Only β-rays are generated. 2: Only Auger & IC electrons are generated.
icharctx=	(D=0) Option for the production of characteristic X-rays. (only for proj=photon) 0: All photons (γ-rays and characteristic X-rays) are generated. 1: Only γ-rays are generated. 2: Only characteristic X-rays are generated.

The following is an example using e-type = 28,29:

```
[ Source ]
 totfact = 2.0
 <source> = 1.0
   s-type =   1
     proj =  photon
      dir =   all
       r0 =   0.
       z0 =   0.
       z1 =   0.
   e-type =   28
       ni =   1
            Cs-137  100.
    dtime =  -10.0
   actlow =   1.0
 <source> = 1.0
   s-type =   1
     proj =  photon
      dir =   all
       r0 =   0.
       z0 =   0.
       z1 =   0.
   e-type =   28
       ni =   1
            Cs-134  100.
    dtime =  -10.0
   actlow =   1.0
```

In this case, Cs-137 and Cs-134 after reaching radioactive equilibrium at 100 Bq are both defined as photon sources. Furthermore, setting actlow=1 ensures that activity smaller than 1 Bq is ignored.

Furthermore, for example, to consider both γ and β-rays by decay of Cs-137, make a multi-sources having two <source> sections with proj=photon and proj=electron. Because both the production rates of the γ and β-rays are the same, their ratios can be specified as 1.0 (<source>=1.0). In this case, totfact should be 2.0, which is the sum of the <source> sections.

Energy distribution of result of tally

Table 5.59: Parameters for source energy distribution (9)

parameter	explanation
`e-type= 20`	Results of tallies can be used as an energy distribution by specifying `file`. Only results of `[t-track]`, `[t-cross]`, `[t-point]`, `[t-product]`, `[t-time]`, and `[t-interact]` tallies with `axis=eng` can be used. A file name (`***.out`), which was specified in the tally section as an output file name, should be set in `file`. There are two kinds of tally results, energy differential and integrated values, in accordance with `unit` given in the tally section. The energy distribution of the source particle is expressed on the basis of the kind. Note that time (nsec) and angular (sr) derivatives are not taken into account. The distribution of the source is normalized to the integrated value of the tally result.
`file=`	Output file name of tallies.

The first result specified by `part` in the tally section is used as the distribution of the source particle. Note that particles of its kind are not generated. The parameter `proj` should be set to specify the kind of the source particle.

An example using `e-type=20` is as follows. This function can be used by setting only `e-type=20` and `file`.

```
[ Source ]
   s-type =   1        # axial source with energy spectrum
     proj =  neutron   # kind of incident particle
      dir =   1.0      # z-direction of beam [cosine]
       r0 =   0.       # radius [cm]
       z0 =   0.       # minimum position of z-axis [cm]
       z1 =   0.       # maximum position of z-axis [cm]
   e-type =   20       # energy distribution given by tally output
     file = cross.out  # file name of tally output
```

5.3.20 Definitions for angular distribution

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

Table 5.60: Parameters for source angular distribution (1)

parameter	explanation
`a-type = 1, (11)`	Any angular distribution can be specified by giving data set of angle bins `a(i)` and integrated values of the particle generation probability `w(i)` by hand. The number of the particle generation in the bin is proportional to `w(i)`, and the specified angular distribution is statistically described. For 1 case, angle is given by cosine, and for 11 case, given by degree.
`na =`	Number of angular group. 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)`	The same angular distribution as in the case of `a-type=1,(11)` can be specified. Unlike `a-type=1,(11)`, the distribution is described by giving data set of angle bins `a(i)` and weights of the source particle `w(i)` by hand. The number of source particles generated in each bin is the same for all angle bin, but integrated values of the weight of source particles are adjusted to be proportional to `w(i)`. The number of source particles generated in each bin can also be changed by specifying `q(i)`. For 4 case, angle is given by cosine, and for 14 case, given by degree.
`na =`	Number of angular group. Data must be given from the next line by the format as `(a(i),w(i),i=1,na), a(na+1)`. In default (`q-type=0`), equal number of particle is generated in each cell. The integrated number of source particles generated in each bin is proportional to `q(i)`.
`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 5.61: Parameters for source angular distribution (2)

parameter	explanation

`a-type = 5, (15)`
`g(x) =`
	Angular distribution `d`φ`/d`Ω`(i)` 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. x denotes angle. One can use intrinsic functions and constants `C`, e.g., `g(x) = exp(-c1*x**2)`.

`nn =`	Number of angular group.

`ag1 =`
`ag2 =`
	minimum cut off for angular distribution.
	maximum cut off for angular distribution.

`a-type = 6, (16)`	The same angular distribution as in the case of `a-type=5,(15)` can be specified. Unlike `a-type=5,(15)`, the number of source particle generated in each bin is the same for all angle bin, but integrated values of the weight of source particles are adjusted to be proportional to $g(x)$. The number of source particles generated in each bin can also be changed by specifying `q(i)`. For 6 case, angle is given by cosine, for 16 case, given by degree.
`g(x) =`	any analytical function of x, Fortran style. One can use intrinsic functions and constants `C`.
`nn =`	Number of angular group. In default (`q-type=0`), equal number of particle is generated in each cell. The integrated number of source particles generated in each bin is proportional to `q(i)`.
`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)`.

5.3.21 Definition for time distribution

For t-type=0,1,2,3,4,5,6, the time distribution function can be set to define the time spectrum of the source particles. If a parameter has a default value (D=***), the parameter can be omitted.

Table 5.62: Parameters for time distribution (1)

parameter	explanation
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 =	width of time distribution [ns]. When t-type=1, it means full width. When t-type=2, it means FWHM of Gaussian distribution.
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]
t-type = 3	Any time distribution can be specified by giving data set of time bin t(i) and integrated values of the particle generation probability w(i), and specified time distribution is statistically described.
ntt =	Number of time group. Data must be given from the next line by the format as (t(i),w(i),i=1,ntt), t(ntt+1). The integrated number of the particle generation in the each time bin is proportional to w(i).
t-type = 4	The same time distribution as in the case of t-type=1 can be specified. Unlike t-type=1, the distribution is described by giving data set of time bins t(i) and weights of the source particle w(i) by hand. The number of source particles generated in each bin is the same for all time bin, but integrated values of the weight of source particles are adjusted to be proportional to w(i). The number of source particles generated in each bin can also be changed by specifying o(i).
ntt =	Number of time group. Data must be given from the next line by the format as (t(i),w(i),i=1,ntt), t(ntt+1). In default (o-type=0), equal number of particle is generated in each cell. The integrated number of source particles generated in each time bin is proportional to o(i).
o-type = 0, 1	(D=0) generation option. For 0, o(i)=1 for all i is assumed without the following data. For 1, o(i) must be given from the next line by the format as (o(i),i=1,ntt).

Table 5.63: Parameters for time distribution (2)

parameter	explanation
`t-type = 5` $h(x)$	Differential spectrum ($d\varphi/dt$) is given by $h(t)$. Any analytical function of x[ns], Fortran style. One can use intrinsic function and constants C.
ll	Number of time group. If it is given by positive number, linear interpolation is assumed in a bin. If negative, logarithmic interpolation is assumed in a bin. Integrated number of source particles generated in each cell is proportional to $h(x)$.
tg1 tg2	Minimum cut off for time distribution [ns]. Maximum cut off for time distribution [ns].
`t-type = 6` $h(x)$	The same time distribution as in the case of `t-type=5` can be specified. Unlike `t-type=5`, the number of source particle generated in each bin is the same for all time bin, but integrated values of the weight of source particles are adjusted to be proportional to $h(x)$. The number of source particles generated in each bin can also be changed by specifying `o(x)`. Any analytical function of x[ns], Fortran style. One can use intrinsic function and constants C.
ll	Number of time 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 (`o-type=0`), equal number of particle is generated in each cell. The integrated number of source particles generated in each time bin is proportional to `o(i)`.
tg1 tg2	Minimum cut off for time distribution [ns]. Maximum cut off for time distribution [ns].
`o-type = 0, 1`	(D=0) generation option. For 0, `o(i)=1` for all `i` is assumed without the following data. For 1, `o(i)` must be given from the next line by the format as `(o(i),i=1,ntt)`.

When setting `t-type=100`, an original distribution can be used by defining its function in subroutine “tdis01” of “sors.f,” which a source file of PHITS.

Table 5.64: Parameters for time distribution (3)

parameter	explanation
`t-type = 100`	User define time distribution. User can write any type of time distribution in the source program of “sors.f”.
tg1 tg2	Minimum cut off for time distribution [ns]. Maximum cut off for time distribution [ns].

5.3.22 Example of multi-source

The following is an example of multi-source, which includes energy distribution and angular distribution described by analytic functions.

Example 4: Example of multi-source

```
[ Source ]
 totfact = 3
<source> = 9.72
  s-type = 1
    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 = 1
    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 = 1
    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, a cylinder source from z =2 cm to z =29 cm with 5 cm radius is defined, and `r1=4` is set. This `r1=4` means that the region inside the cylinder with radius 4 cm is not included. In the next source, it is also a cylinder source from z =1 cm to z =2 cm with 5 cm radius without `r1`. This is a normal thin cylinder. The last one is also a thin cylinder from z =29 to z =30 cm with 5 cm 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. 5.6 using `[t-product]` tally with `output=source`, and `icntl=6`.

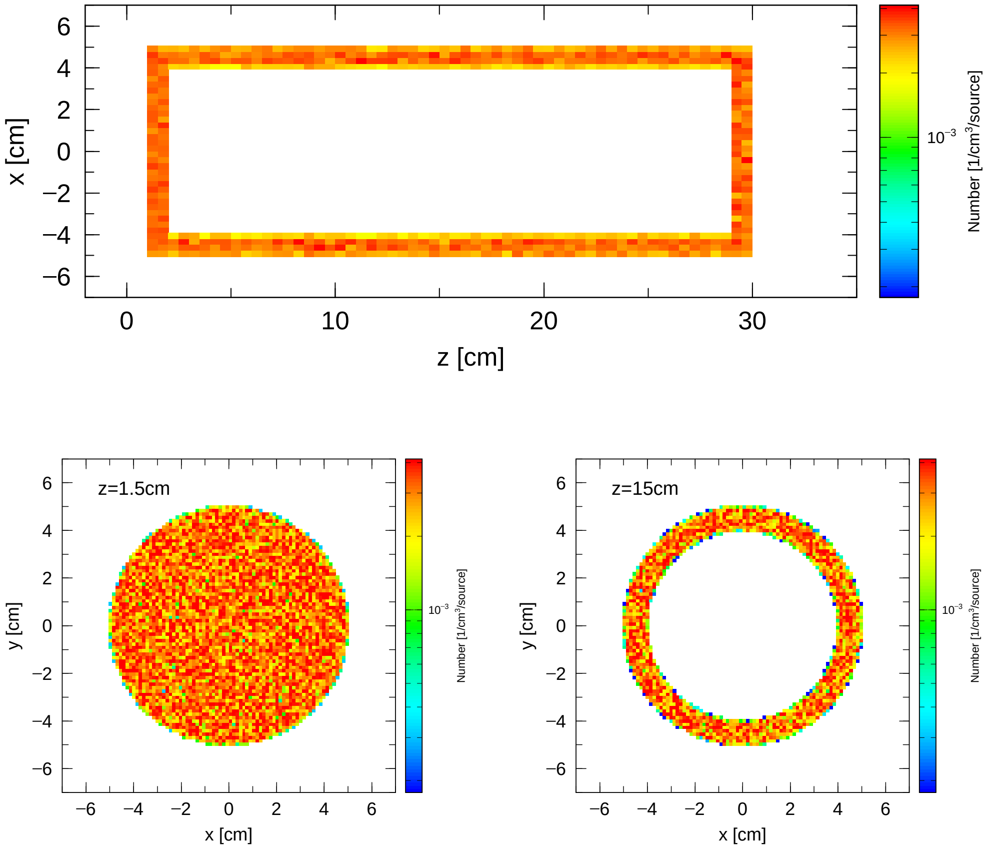


Figure 5.6: Multi-source, coordinate distribution

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. 5.7 with different colors.

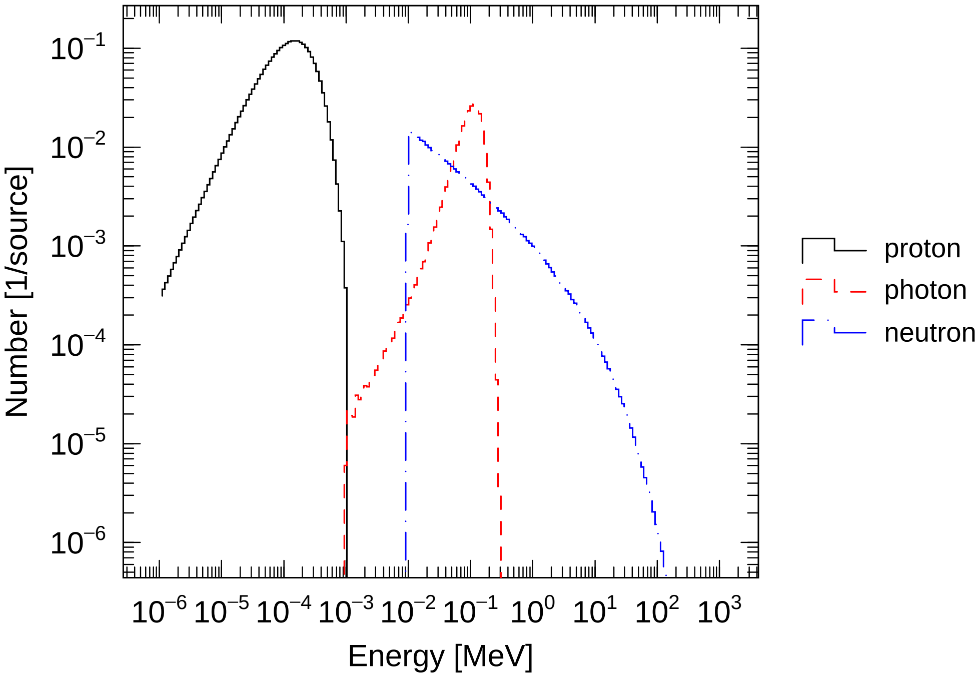


Figure 5.7: Multi-source, energy distribution

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 an analytic function is used for an angular distribution. The angular distribution of the third one is shown in Fig. 5.8 by using `[t-cross]` tally.

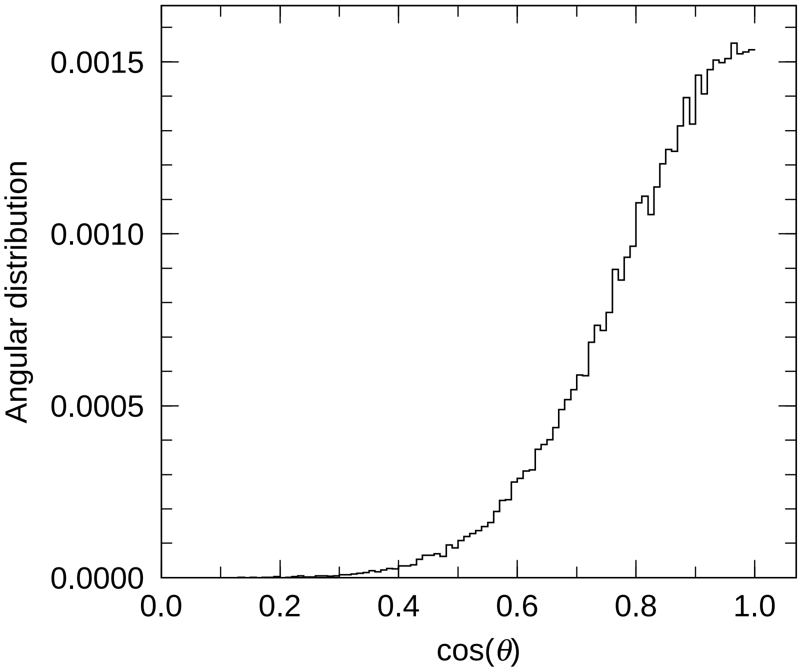


Figure 5.8: Multi-source, angular distribution

5.3.23 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, the following duct source options can be used 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 an isotropic distribution of the source direction is assumed. To reduce this variance, a special option of the source function cab be used. When using this option, 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.

The duct source option is available when setting `dom=-10` in the circle or rectangle source, which are set by `s-type=1` or `2`, respectively. The parameters for the duct source options are summarized in Table 5.65.

Table 5.65: 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[cm].
`dxw =`	x size of rectangle beam line for s-type = 2[cm].
`dyw =`	y size of rectangle beam line for s-type = 2[cm].

A shape of the beam-line is assumed to be circle or rectangle for `s-type=1` or `2`, respectively. The setting of `z1=z0` and `dir=1` is also assumed. The latter parameter `dir` means the direction of the beam-line. To change the direction of the beam-line, use the transformation by setting `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` or `2`, 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`. If the source region at `z0` can be seen at all duct wall positions from `dl0` to `dl2`, the normalization factor is set to be 1. 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, the angular distribution of the source particle is assumed to be isotropic within the small solid angle which covers the whole beam-line.

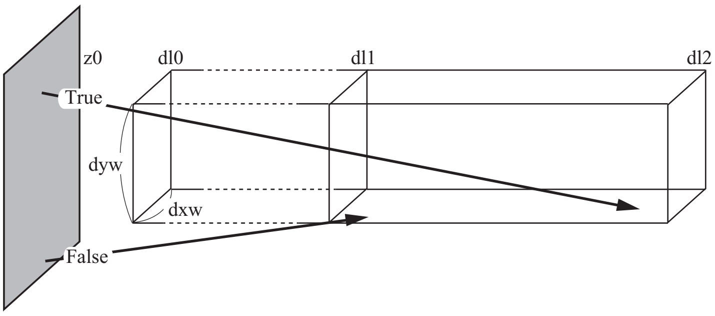


Figure 5.9: Schematic image of the duct source.

Some example of the duct source option are shown below.

In the first example, the rectangle source and beam-line, the same size of the source and beam-line dimensions, are defined. The input for the duct source option is as follows:

Example 5: duct source option, example (1)

```
[ Source ]

set: c1[200]    $dl0
set: c2[500]    $dl1
set: c3[5000]   $dl2
set: c4[5.0]    $x*2 at z0
set: c5[5.0]    $y*2 at z0
set: c10[5.0]   $dxw
set: c20[5.0]   $dyw
set: c30[0.001] $dpf

s-type = 2
proj   =  neutron
e0     =  20.0
x0     = -c4/2
x1     =  c4/2
y0     = -c5/2
y1     =  c5/2
z0     =  0.0
z1     =  0.0
dir    =  1.0
phi    =  0.0

dom = -10
dl0 = c1
dl1 = c2
dl2 = c3
dxw = c10
dyw = c20
dpf = c30
```

In the first part of above source section, some constants are set to define the duct source option, dl0, dl1, dl2, and size of source, dxw, dyw, dpf. In the second part, the position and xy region of the source, direction of the beam-line and the energy of source particle are defined. In the third part, the duct source options are defined. The setting gives the calculaton of the 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. 5.10 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.

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

Example 6: duct source option, example (2)

```
[ Source ]

set: c1[200]    $dl0
set: c2[500]    $dl1
set: c3[5000]   $dl2
set: c4[10.0]   $x*2 at z0
set: c5[10.0]   $y*2 at z0
set: c10[5.0]   $dxw
set: c20[5.0]   $dyw
set: c30[0.001] $dpf

s-type = 2
proj   =  neutron
e0     =  20.0
x0     = -c4/2
x1     =  c4/2
```

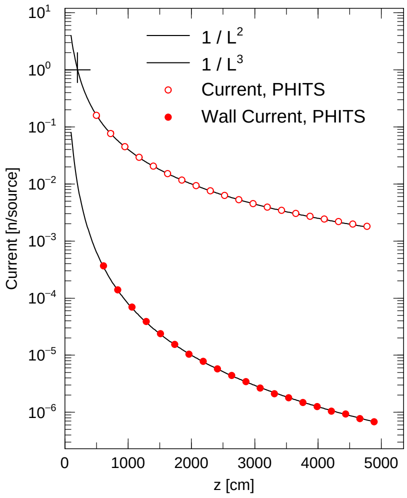


Figure 5.10: Duct source option, example 1

```
y0     = -c5/2
y1     =  c5/2
z0     =  0.0
z1     =  0.0
dir    =  1.0
phi    =  0.0

dom = -10
dl0 = c1
dl1 = c2
dl2 = c3
dxw = c10
dyw = c20
dpf = c30
```

Figure 5.11 shows how the extra region of the source increases the current and the wall current. By this function, the margin area of the moderator to the size of the cross-section of beam-line can be automatically treated.

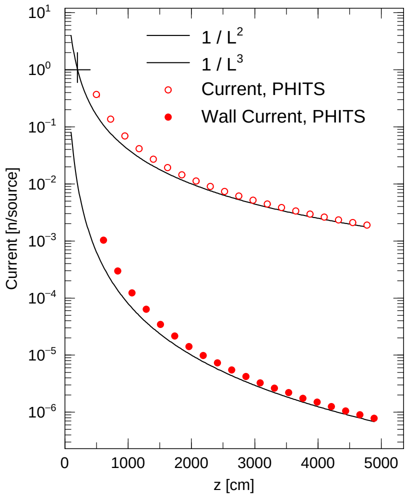


Figure 5.11: Duct source option, example 2

5.4 [Material] section

5.4.1 Formats

Materials, which are used to define 3-dimensional geometry, are defined in this section. The defined materials are refered to some sections such as `[cell]` section.

The `[material]` section is defined by material number, elements (or nuclides), and their composition ratios. Following a material number, element symbols and their ratios should be alternately written to define a compound or mixture. The format is:

```
[ Material ]
MAT[n]     element   ratio     element   ratio     ...     ...
```

Two formats to specify material numbers can be used as follows: `MAT[`*n*`]` and `M`*n*, where *n* can be specified up to a material number of 99,999 unless it is over-defined. Note that a blank space cannot be set between `MAT` and `[` in the format `MAT[`*n*`]`.

The following comment marks can be used: `#, %, !, $`. Although version 2.88 or before, `c` can be also used as comment marks, after version 2.89, it is not permitted in the default setting. To use `c` as comment marks in this section, set `icommat=1` in the `[parameters]` section. Note that the `[parameters]` section with `icommat=1` should be written above (before) the `[material]` section including `c`.

5.4.2 Element (nuclide) definition

The `element` in the above format can be specified using element symbols. To define nuclide (isotope), add its mass number to the symbol as follows: `208Pb, Pb-208, or 82208`. For instance, hydrogen is defined as `1H, H-1, or 1001`. The natural abundance of isotopes can be defined using only an element symbol or a no-mass style as, e.g., Pb, 82000. Note that the natural isotope ratio format cannot be used for an isotope not included in the nuclear data library JENDL-4.0.

To use nuclear data libraries other than JENDL-4.0, copy all addresses of the new library and then paste it at the end of the address file `file(7) (D=xsdir)`. When setting the new library of low-energy neutrons (below 20MeV) other than JENDL-4.0, specify the "library id" in each material number by `NLIB` parameter, which is shown in Table 5.66. The "library id" consists of the library number (double-digit) and data class (character a-z). For example, `50c` for the low-energy neutron data library of JENDL-4.0. The libraries of photons, electrons, and protons can be also specified by `PLIB, ELIB, HLIB`, respectively. The library id can be specified by nuclide instead of by material number following the element definition and a period as, e.g., `208Pb.50c, Pb-208.50c, or 82208.50c`. In PHITS, only one type of the library should be used for each incident particle, and the id should be specified by the `NLIB, PLIB, ELIB, HLIB` parameters. Only for neutrons, the second library can be used; in this case, after specifying the library id by `NLIB`, define the nuclide by the expression of `208Pb.50c`. If the library id is not specified by the user, PHITS searches the address file `file(7) (D = xsdir)` from the top line for the library number id corresponding to the nuclide and uses the corresponding data library. Information on the data library used in a PHITS calculation is written in the summary output file `file(6) (D=phits.out)` when `kmout=1` is set in the `[parameters]` section.

When using EGS5, restrict the number of elements (not nuclides) in one material number to 20.

5.4.3 Composition ratio definition

The composition ratios of the defined elements are given in `ratio` in the above format. Two ways are available for the ratio definition: if ratio takes a positive value, it means atomic ratio; for negative value, it does mass ratio. For instance, water (H_2O) can be defined as follows:

```
MAT[1] H 2  O 1
```

or

```
MAT[1] H -2/18  O -16/18
```

In the latter case, the mass ratios of hydrogen and oxygen are given as about 2/18 and 16/18, respectively, because the molecular mass of the water is about 18.

The material densities used in transport calculations are given in the [cell] section. Note that material densities must be defined in the [material] section instead of composition ratios when using letmat parameter in [t-deposit], [t-deposit2], [t-let], [t-sed] sections. In this case, if it takes a positive value, the particle density is [10^{24} atoms/cm^3], and for negative value, it is given as a mass density [g/cm^3].

5.4.4 Material parameters

For regions in which nuclear data are used, the material parameters for each material can be set using the format *keyword=value*. The parameters can be set anywhere in the material subsection. The full set of material parameters is listed in Table 5.66.

Table 5.66: Material parameters

parameter	value	explanation
GAS	D = 0	Density effect correction to electron stopping power.
	= 0	Appropriate for materials in the condensed (solid or liquid) state used.
	= 1	Appropriate for material in the gaseous state.
ESTEP	= *n*	Make sub step number *n* for electron transport: ignored when *n* is smaller than the 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 exists at least 1 non-conductor; otherwise, conductor.
	> 0	Conductor if there exists at least 1 conductor.

5.4.5 $S(\alpha,\beta)$ settings

In the transport of low-energy neutrons, the library of the thermal scattering law data $S(\alpha,\beta)$ may be required. This library plays an important role in describing the transport of thermal neutrons. This library can be set as follows:

```
MTn materialID
```

where n is the material number and materialID is the ID number, such as lwtr.20t, written in “xsdir”. For example, the library for water at room temperature (at 296K) can be set as follows:

```
M1   H    2.0
     O    1.0
MT1  lwtr.20t
```

See “\phits\XS\tsl\tsl-table” for detailed information for these data.

5.4.6 Examples

Some examples using the materials parameter are shown below.

Example 7: material example (1)

```
[ Material ]
MAT[ 1 ]
   1H        1.0000000E-04
 208Pb       1.7238000E-02
 204Pb       4.6801000E-04
 206Pb       7.9430000E-03
 207Pb       7.2838000E-03
MAT[ 2 ]
   1H        1.0000000E-09
  14N        4.6801000E-05
  16O        7.9430000E-06
```

By default, the order is element, then ratio; these can be specified in reverse by putting `den` and `nuc` as,

Example 8: material example (2)

```
[ Material ]
   den            nuc      <------
MAT[ 1 ]
1.0000000E-04      1H
1.7238000E-02    208Pb
4.6801000E-04    204Pb
7.9430000E-03    206Pb
7.2838000E-03    207Pb
MAT[ 2 ]
1.0000000E-09      1H
4.6801000E-05     14N
7.9430000E-06     16O
```

Example 9: material example (3)

```
[ Material ]
m1     80196.49c  5.9595d-5
       80198.49c  3.9611d-3
       80199.49c  6.7025d-3
       80200.49c  9.1776d-3
       80201.49c  5.2364d-3
       80202.49c  1.1863d-2
       80204.49c  2.2795d-3
c      ...Be...
m3     4009.37c  1.2362E-1
mt3    be.01
c      ...h2o (25C)...
m4     1001.37c  6.6658d-2      8016.37c  3.3329d-2
mt4    lwtr.01
c      ...b4c (natural boron; 25%-density)...
m5     6012.37c  6.8118d-3
       5011.37c  2.1825d-2
c      ...liquid-h2 (20K)...
m6     1001.49c  3.1371d-2      1011.49c  1.0457d-2
mt6    orthoh.00   parah.00
```

5.5 [Surface] section

5.5.1 Formats

In the [surface] section, surfaces can be set to define cells in the [cell] section. Several surfaces, including planes, sphere surfaces, cylindrical surfaces, etc., can be set in terms of the parameters in the equations defining their respective shapes. Only C and $ can be used as comment marks in [surface]. To use continuation lines, at least five blanks can be placed at the line head in place of the line sequential mark at the end of line.

The [surface] section is defined in the order data: surface number, coordinate transform number, surface symbol, and surface definition; these are explained in Table 5.67. The format is as follows:

```
[ Surface ]
surface number     transform number     surface symbol     surface definition
```

The surfaces expressed by the equations in Table 5.68 or the macro body in Table 5.68 can be referred to using their surface symbols. In addition, mathematical expressions and user defined variables can be used for surface definition. Reflective and white boundary conditions can be set by writing * and +, respectively, before the surface number. For example, *10 indicates that surface 10 is a reflective boundary. The reflective boundary condition is useful to develop an infinite repeated structure.

Table 5.67: Surface definition format

item	explanation
surface number	Use any number from 1 to 999,999.
transform number	Specifies the number *n* of TR*n* defined in the [transform] section. Does not specify if not using the coordinate transform.
surface symbol	Specifies a surface symbol in Tables 5.68 and 5.69.
surface definition	Species parameters of 1 ∼ 15 inputs, which depend on the surface symbol.

Table 5.68: Surfaces expressed by equations

Surface Symbol	Type	Explanation	Equation	Parameters
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
P	plane	defined by 3 coordinates		$x_1, y_1, z_1,\ x_2, y_2, z_2,\ x_3, y_3, z_3$
SO	sphere	origin is center	$x^2 + y^2 + z^2 - R^2 = 0$	R
S		multi-purpose	$(x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2 - R^2 = 0$	$x_0,\ y_0,\ z_0,\ R$
SX		center on X-axis	$(x - x_0)^2 + y^2 + z^2 - R^2 = 0$	$x_0,\ R$
SY		center on Y-axis	$x^2 + (y - y_0)^2 + z^2 - R^2 = 0$	$y_0,\ R$
SZ		center on Z-axis	$x^2 + y^2 + (z - z_0)^2 - R^2 = 0$	$z_0,\ R$
C/X	cylinder	parallel with X-axis	$(y - y_0)^2 + (z - z_0)^2 - R^2 = 0$	$y_0,\ z_0,\ R$
C/Y		parallel with Y-axis	$(x - x_0)^2 + (z - z_0)^2 - R^2 = 0$	$x_0,\ z_0,\ R$
C/Z		parallel with Z-axis	$(x - x_0)^2 + (y - y_0)^2 - R^2 = 0$	$x_0,\ y_0,\ R$
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	$\sqrt{(y - y_0)^2 + (z - z_0)^2} \mp \lvert t\rvert\,(x - x_0) = 0$	$x_0,\ y_0,\ z_0,\ \lvert t\rvert^2,\ k$
K/Y		parallel with Y-axis	$\sqrt{(x - x_0)^2 + (z - z_0)^2} \mp \lvert t\rvert\,(y - y_0) = 0$	$x_0,\ y_0,\ z_0,\ \lvert t\rvert^2,\ k$
K/Z		parallel with Z-axis	$\sqrt{(x - x_0)^2 + (y - y_0)^2} \mp \lvert t\rvert\,(z - z_0) = 0$	$x_0,\ y_0,\ z_0,\ \lvert t\rvert^2,\ k$
KX		on X-axis	$\sqrt{y^2 + z^2} \mp \lvert t\rvert\,(x - x_0) = 0$	$x_0,\ \lvert t\rvert^2,\ k$
KY		on Y-axis	$\sqrt{x^2 + z^2} \mp \lvert t\rvert\,(y - y_0) = 0$	$y_0,\ \lvert t\rvert^2,\ k$
KZ		on Z-axis	$\sqrt{x^2 + y^2} \mp \lvert t\rvert\,(z - z_0) = 0$	$z_0,\ \lvert t\rvert^2,\ k$
				k is ±1 or unspecified
SQ	ellipse, hyperboloid, paraboloid	parallel with X-, Y-, or Z- axis	$A(x - x_0)^2 + B(y - y_0)^2 + C(z - z_0)^2$ $+2D(x - x_0) + 2E(y - y_0) + 2F(z - z_0)$ $+G = 0$	$A,\ B,\ C,\ D,\ E,\ F,\ G,\ x_0,\ y_0,\ z_0$
GQ	cylinder, cone, 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	$(x - x_0)^2/B^2+$ $(\sqrt{(y - y_0)^2 + (z - z_0)^2} - A)^2/C^2 - 1 = 0$	$x_0,\ y_0,\ z_0,\ A,\ B,\ C$
TY			$(y - y_0)^2/B^2+$ $(\sqrt{(x - x_0)^2 + (z - z_0)^2} - A)^2/C^2 - 1 = 0$	$x_0,\ y_0,\ z_0,\ A,\ B,\ C$
TZ			$(z - z_0)^2/B^2+$ $(\sqrt{(x - x_0)^2 + (y - y_0)^2} - A)^2/C^2 - 1 = 0$	$x_0,\ y_0,\ z_0,\ A,\ B,\ C$

When using the surfaces defined in the `[surface]` section in the `[cell]` section, the interior of the shape should essentially be specified in a 'negative sense,' while the exterior should be specified in a 'positive sense.' Note that, in the case of unclosed shapes such as planes, it is necessary to distinguish which side of the plane is 'positive' and which is 'negative' by calculating a value of $f(x, y, z)$ at a point (x_0, y_0, z_0), where $f(x, y, z)$ is the equation of the unclosed shape. If $f(x_0, y_0, z_0) > 0$, the side including the point is 'positive'; if $f(x_0, y_0, z_0) < 0$, the side is 'negative.' For example, for shape symbol PY with $D = 5$, the side including the origin $(0, 0, 0)$ is 'negative' because $f(0, 0, 0) = -5 < 0$.

A plane can be defined by assigning the symbol `P` to the x, y, z-coordinates of three points. In this case, the region including the origin is a negative region.

A cone can be defined in terms of its vertex point x_0, y_0, and z_0 and aligned with any of the x, y, or z axes. By default, the cone will have two sheets extending from the vertex (see Example 14); If parameter k is set = 1, the upper sheet is used, while the lower sheet is used when $k = -1$. If k is not specified, both sheets are used. Note that if a region is defined in the `[cell]` section using only one cone sheet, the plane passing through the vertex must be specified. In defining a cone, three surfaces —the vertex-crossing plane, the side sheet of the cone, and the underside of the cone— must be specified.

Note that when using `TX,TY,TZ` to define an elliptical torus or torus, the shape's coordinates cannot be transformed by setting the `[transform]` section.

Table 5.69: Macro body

Symbol	Type	Parameters	Explanation
BOX	Optional BOX (all angles are 90°)	x_0, y_0, z_0, A_x, A_y, A_z, B_x, B_y, B_z, C_x, C_y, C_z	Base point coordinate Vector from base point to first surface. Vector from base point to second surface. Vector from base point to third surface.
RPP	Rectangular solid (Each surface is vertical with x, y, z axes)	$x_{\min}$, $x_{\max}$, $y_{\min}$, $y_{\max}$, $z_{\min}$, $z_{\max}$	Minimum and Maximum of x. Minimum and Maximum of y. Minimum and Maximum of z.
SPH	Sphere (the same with general sphere S)	x_0, y_0, z_0, R	Center coordinate. Radius.
RCC	Cylinder	x_0, y_0, z_0, H_x, H_y, H_z, R	Center coordinate of bottom face. Height vector from center of bottom face to that of top face. Radius.
RHP or HEX	Optional hexagonal prism Prism	x_0, y_0, z_0, H_x, H_y, H_z, A_x, A_y, A_z, B_x, B_y, B_z, C_x, C_y, C_z	Base point coordinate. Height vector from base point. Vector from base point to first surface. Vector from base point to second surface. Vector from base point to third surface.
REC	Right elliptical cylinder	x_0 y_0 z_0, H_x, H_y, H_z, A_x, A_y, A_z, B_x, B_y, B_z	Center coordinate of bottom face. Height vector from center of bottom face (***H***). Major axis vector of ellipse orthogonal to ***H*** (***A***). Minor axis vector of ellipse orthogonal to ***H*** and ***A*** (***B***).
TRC	Truncated right-angle cone	x_0, y_0, z_0, H_x, H_y, H_z, R_1, R_2	Center coordinate of bottom face of cone. Height vector from center of bottom. Radius of bottom face. Radius of top face.
ELL	Ellipsoid of revolution (Spheroid)	If $R > 0$, x_1, y_1, z_1, x_2, y_2, z_2, R If $R < 0$, x_0, y_0, z_0, A_x, A_y, A_z, R	 Coordinate of first focus. Coordinate of second focus. Radius of major axis. In this case, the ellipsoid formed by a rotation on the major axis. The major axis can be set to be shorter than the minor axis. Center coordinate of ellipsoid. Major axis vector. Radius of minor axis.
WED	Wedge	x_0, y_0, z_0, A_x, A_y, A_z, B_x, B_y, B_z, H_x, H_y, H_z	Coordinate of top. Vector to first side of triangle (***A***). Vector to second side of triangle (***B***). Height vector (***H***).

Note that, in defining `TRC`, R_2 cannot be set to 0. To define a typical cone, i.e., one that is not truncated, using `TRC`, set R_2 to a small value.

The bottom of a wedge defined by `WED` must be a right triangle. To make an arbitrary triangle, right triangles of various sizes must be combined.

5.5.2 Examples

This section shows examples of the surface definitions listed in Tables 5.68 and 5.69.

Example 10: Example of Plane (1)

```
[surface]
  1  PY    5
```

PX,PY,PZ define planes perpendicular to the x, y, z axes, respectively. In this example, a plane of surface number 1, corresponding to $y = 5$ and perpendicular to y axis, is defined by setting the symbol to PY and the parameter to $D = 5$. Surface number 1 corresponds to the plane shown as a dashed line in Fig. 5.12.

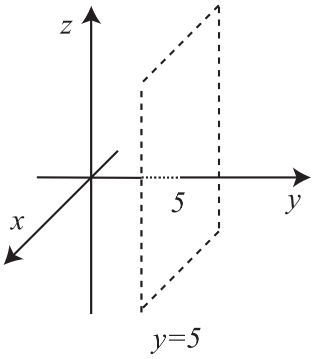


Figure 5.12: The plane $y = 5$ defined in Example 10.

Example 11: Example of Plane (2)

```
[surface]
  1  P     2 2 1  10
  2  P     5 0 0  0 5 0  0 0 10
```

Arbitrary planes can be defined by specifying P as the symbol. In this case, two types of description can be used to define the plane: four parameters, A, B, C, D, can be used to specify the plane equation $Ax + By + Cz - D = 0$; otherwise, the coordinates of three points, $(x_1, y_1, z_1), (x_2, y_2, z_2), (x_3, y_3, z_3)$, can be specified. The former method is useful when the normal vector (s, t, u) and coordinates (x_0, y_0, z_0) of one point on the plane are known. An equation of the plane satisfying these conditions would be given as

$$s(x - x_0) + t(y - y_0) + u(z - z_0) = 0. \tag{6}$$

Then, because

$$sx + ty + uz - (sx_0 + ty_0 + uz_0) = 0, \tag{7}$$

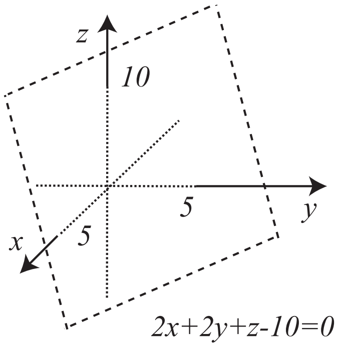


Figure 5.13: The plane defined in Example 11.

the plane can be defined by setting $A = s, B = t, C = u, D = sx_0 + ty_0 + uz_0$. The surface number 1 shown in Example 11 is defined as a plane having the normal vector $(2, 2, 1)$ and containing the coordinate $(5, 0, 0)$. This surface corresponds to the plane denoted by the dashed line in Fig. 5.13. As an illustration of the latter method, surface number 2 is defined as a plane passing through the three coordinates $(5, 0, 0), (0, 5, 0), (0, 0, 10)$. This surface coincides with surface 1 in Figure 5.13.

Example 12: Example of Sphere Surface

```
1:   [surface]
2:     1   SO     5
3:     2   SZ     10  3
4:     3   S      10 10 0  3
```

To define a sphere surface, the symbols `SO,SX,SY,SZ,S` can be used. `SO` is used to define a sphere surface centered at the origin by specifying its radius R as a parameter. In the second line of Example 12, a sphere surface centered at the origin with a radius of 5 cm is defined as surface number 1. The symbols `SX,SY,SZ` can be used to define sphere surfaces centered on the x, y, z axes, respectively. To use these symbols, two parameters —a coordinate value on the appropriate axis, and the radius of the sphere— must be specified. In the third line of Example 12, a sphere surface centered at the coordinate $(0, 0, 10)$, which is a point on the z axis, with a radius of 3 cm is defined as surface number 2. `S` can be used to define a sphere surface centered at an arbitrary coordinate (x_0, y_0, z_0). In this case, four parameters —the coordinate values x_0, y_0, z_0 and the radius R— must be specified. In the fourth line of Example 12, a sphere surface centered at the coordinate $(10, 10, 0)$ with a radius of 3 cm is defined as surface number 3. The three sphere surfaces defined in Example 12 are shown in Fig. 5.14.

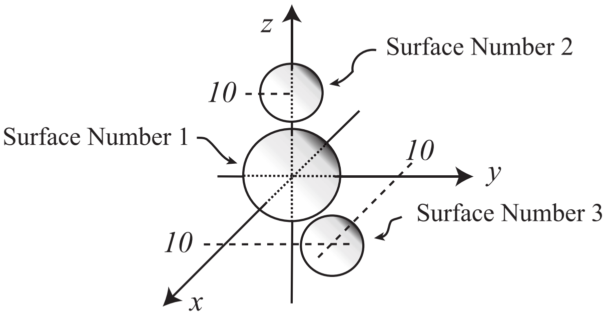


Figure 5.14: The three sphere surfaces defined in Example 12.

Example 13: Example of Cylinder Surface

```
1:   [surface]
2:     1   CY     5
3:     2   C/Y    15 0  3
```

To define a cylinder surface having a central axis along either the x, y, z axes, the symbols `CX,CY,CZ`, respectively, can be specified. In this case, the radius R must be specified as a parameter. In the second line of Example 13, a cylinder surface with a central axis on the y axis with a radius of 5 cm is defined. This surface corresponds to the surface of the cylinder shown in the upper-right of Fig. 5.15. The symbols `C/X,C/Y,C/Z`, can be used to define a cylinder surface with a central axis parallel to the x, y, z axes, respectively. In this case, two coordinate values and the radius of the cylinder must be specified. The third line of Example 13 defines a cylinder surface with a central axis parallel to the y axis. The coordinates $x_0 = 15, z_0 = 0$ are specified as the first and second parameters, respectively, and the radius is given by the third parameter. This surface corresponds to the cylinder in the lower-left side of Fig. 5.15.

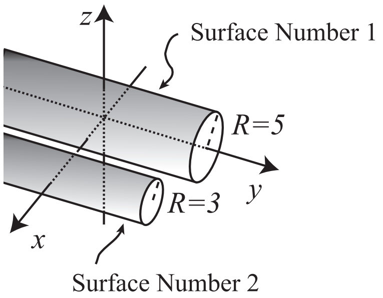


Figure 5.15: The two cylinder surfaces defined in Example 13.

To define a cylinder surface with a central axis that is NOT parallel to any of the x, y, z axes, the symbol `GQ` can be used. However, it is easier to use `CX,CY,CZ,C/X,C/Y,C/Z` and then perform a coordinate transformation using the `[transform]` section than it is to use `GQ`.

Example 14: Example of Cone Surface

```
1:   [surface]
2:     1   KZ     0  1
3:     2   K/Z    0 20 0  1/3  1
```

A cone surface with a central axis along the x, y, z axes can be defined by specifying the symbols `KX,KY,KZ`, respectively. Parameters defining the coordinate of the vertex, an angle parameter, $|t|^2$, and a sheet parameter, k (omitted), must also be set. An example in which `KZ` is specified is shown in the second line of Example 14. The first parameter is the z-coordinate of the vertex of the cone, $z_0 = 0$, and the second defines $|t|^2 = 1$. The equation of `KZ` is

$$\sqrt{x^2 + y^2} \mp |t|\,(z - z_0) = 0. \tag{8}$$

After substituting each of the parameters, the following equations are obtained:

$$\sqrt{x^2 + y^2} - z = 0, \ (z > 0) \tag{9}$$

$$\sqrt{x^2 + y^2} + z = 0, \ (z < 0) \tag{10}$$

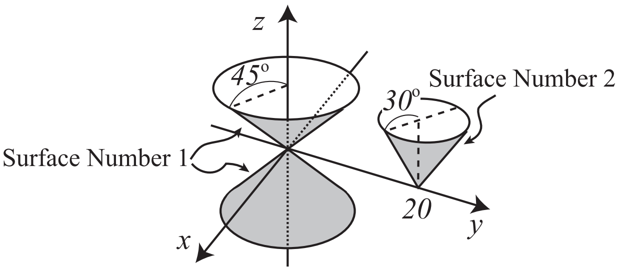


Figure 5.16: The cone surface defined in Example 14.

This formulation corresponds to surface number 1, which is defined as the cone surface having a central axis on the z axis with its vertex at $(0, 0, 0)$. This surface is shown on the left side of Fig. 5.16. The cone in the region $z > 0$ is represented by Eq. (9), while that in $z < 0$ is given by Eq. (10); these surfaces are specified by $k = 1$ and $k = 1$, respectively. Note that, as k is not specified in the second line of the example, both surfaces are defined. The parameter $|t|^2$ is given by the relation $|t|^2 = \tan^2\theta$, where θ is the angle between the central axis and the generating line. In the case of surface number 1, $\tan\theta = 1$, and therefore $\theta = 45°$. To define a cone surface with a central axis parallel to either the x, y, z axes and a vertex at (x_0, y_0, z_0), the symbols `K/X,K/Y,K/Z`, respectively, can be used. The third line of Example 14 shows an example in which `K/Z` is specified. The coordinate of the vertex is $(0, 20, 0)$, and the cone has a central axis parallel to the z axis. Its angle θ is $= 30°$, corresponding to $|t|^2 = 1/3$, or $\tan\theta = 1/\sqrt{3}$. In this case, only one cone surface in the region of $z > 0$ is defined by specifying $k = 1$.

Example 15: Example of Ellipsoid Surface

```
1:   [surface]
2:     1   SQ     1/9**2 1/6**2 1/3**2  0 0 0  -1  0 0 0
```

To define an ellipsoid surface as shown in Fig. 5.17, the symbol `SQ` is specified. The general equation of an ellipsoid surfaces is

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} + \frac{z^2}{c^2} = 1 \tag{11}$$

where a, b, c are the lengths of the semi-principal axes along the x, y, z axes, respectively. An ellipsoid surface can therefore be defined by specifying the first to the seventh parameters of `SQ` as $A = 1/a^2, B = 1/b^2, C = 1/c^2$, $D = E = F = 0$, and $G = -1$, respectively. In Example 15, an ellipsoid surface with lengths $a = 9, b = 6, c = 3$ cm is defined. To define a spheroid surface, the two lengths along the two axes that are not the rotational axis must have same values. For example, if the x axis is the rotational axis, b must be set $= c$, namely, $B = C$. The eighth, ninth, and tenth parameters of `SQ` correspond to the coordinates of the center (x_0, y_0, z_0, respectively). In Example 15, the center of the ellipsoid is the origin $(0, 0, 0)$. Note that if D, E, F are set to non-zero values, a quadric surface other than an ellipsoid is defined.

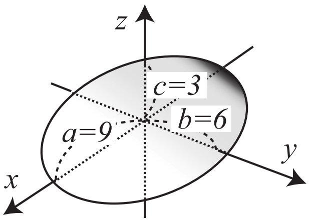


Figure 5.17: The ellipsoid surface defined in Example 15.

Example 16: Example of Hyperboloid Surface

```
1:   [surface]
2:     1   SQ     1/6**2 1/3**2 -1/5**2  0 0 0  -1  0 0 0
3:     2   SQ     -1/6**2 -1/3**2 1/5**2  0 0 0  -1  0 20 0
```

The symbol SQ can be specified to define hyperboloid surfaces, as shown in the center and right-hand side of Fig. 5.18. The surface in the center of the figure is called the hyperboloid of one sheet; those on the right-hand side are hyperboloids of two sheets. Equations of hyperboloids of one and two sheets are given as, respectively,

$$\frac{x^2}{a^2}+\frac{y^2}{b^2}-\frac{z^2}{c^2} = 1, \quad (12)$$

$$-\frac{x^2}{a^2}-\frac{y^2}{b^2}+\frac{z^2}{c^2} = 1. \quad (13)$$

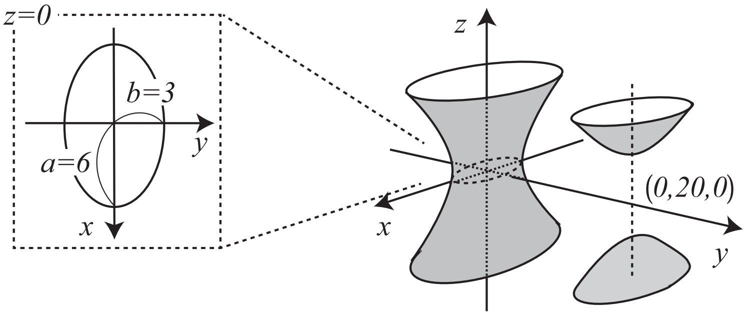


Figure 5.18: The hyperboloid surface defined in Example 16.

Here, the central axis is the z axis. If $a = b$, the shape is called a hyperboloid of revolution. The surface number 1 defined in the second line of Example 16 corresponds to the hyperboloid of one sheet shown in the center of Fig. 5.18. In the second line, the first to the seventh parameters of SQ are given as $A = 1/6^2, B = 1/3^2, C = -1/5^2$, $D = E = F = 0$, and $G = -1$, respectively. The central axis, which corresponds to the z axis, passes through the origin because the eighth to tenth parameters are set to $x_0 = 0, y_0 = 0, z_0 = 0$, respectively. The intersection of this hyperboloid with a plane vertical to the z axis is an ellipse. The figure in the left side of Fig. 5.18 shows the intersection line of surface number 1 with the plane $z = 0$. This ellipse has the major and minor radii of $a = 6$ cm and $b = 3$ cm, respectively. The surface defined in the third line of Example 16 corresponds to the hyperboloid of two sheets shown in the right-hand side of Fig. 5.18. In this case, the signs of a, b, c of surface number 1 are reversed, and $D = E = F = 0$ and $G = -1$ are specified to define a hyperboloid of two sheets. Note that the central axis passes through the coordinate $(0, 20, 0)$ as (x_0, y_0, z_0) is set = $(0, 20, 0)$ in the eighth, ninth, and tenth parameters, respectively. When a surface of a hyperboloid of two sheets is used to define a cell in the [cell] section, one region of the coordinate system (x_0, y_0, z_0) represents the 'negative sense,' while the remaining space represents the 'positive sense.' The interiors of the two sheets shown in the right-hand side of Fig. 5.18 should be specified as 'positive.'

Example 17: Example of Paraboloid Surface

```
1:   [surface]
2:     1   SQ     1 1/2**2 0  0 0 -1  0  0 0 0
```

To define a paraboloid surface as shown in Fig. 5.19, the symbol SQ can be specified as described in Example 17. The equation of a paraboloid surface with a central axis coinciding with the z axis is given as

$$z = \frac{x^2}{a^2}+\frac{y^2}{b^2}. \quad (14)$$

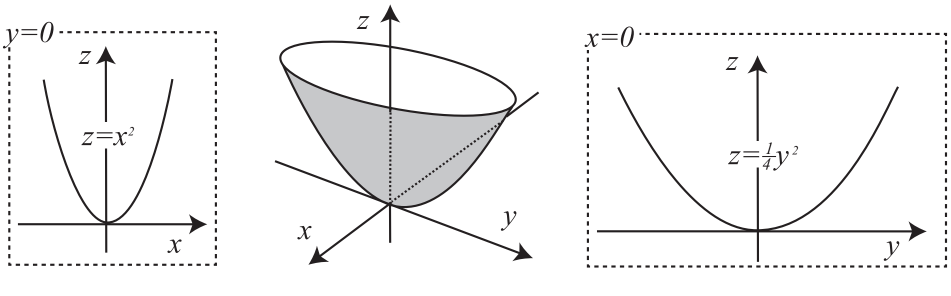


Figure 5.19: The paraboloid surface defined in Example 17.

This surface can be defined by specifying $A = 1/a^2, B = 1/b^2, C = 0$, $D = E = 0, F = -1$, and $G = 0$ as the first to seventh parameters, respectively, of SQ. In the example, a paraboloid with $a = 1, b = 2$ is defined. The intersection of this paraboloid with a plane including the z axis is a parabola. The intersection lines of the paraboloid with $y = 0$ and $x = 0$ are shown in the left and right panels of Fig. 5.19, respectively. The coordinates of the vertex are specified by eighth through tenth parameters; in this case, $x_0 = 0, y_0 = 0, z_0 = 0$.

Example 18: Example of `GQ`

```
[surface]
set: c1[30]
set: c2[cos(c1/180*pi)]
set: c3[sin(c1/180*pi)]
  1   GQ     c2**2 1/2**2 c3**2  0 0 -2*c2*c3  -c3 0 -c2  0
```

Arbitrary surfaces expressed by quadratic equations in x, y, z by can be defined specifying the symbol `GQ`. Although there similar surfaces such as `CX,KX,SQ` are also expressed by quadratic equations, `GQ` can be used to represent quadratic surfaces with a central axis that is NOT parallel to the x, y, z axes. Note that it is easier to use apply a coordinate transformation in the `[transform]` section to `CX,KX,SQ` than it is to use `GQ`. Example 18 represents a paraboloid surface obtained by rotating the surface in Example 17 by an angle of 30° around the y axis. The result is shown in Fig. 5.20. The parameters of this example were obtained by calculating the following transform defined by rotating the coordinate (x, y, z) by an angle of θ around y axis:

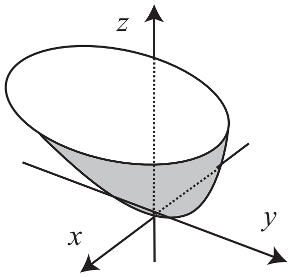


Figure 5.20: The paraboloid surface defined in Example 18.

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix}. \tag{15}$$

This relation gives

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{pmatrix} \begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} x'\cos\theta - z'\sin\theta \\ y' \\ x'\sin\theta + z'\cos\theta \end{pmatrix}. \tag{16}$$

After substitution of this result into Eq. (14), the following equation of x', y', z' is obtained:

$$\begin{aligned}
x'\sin\theta + z'\cos\theta &= \frac{(x'\cos\theta - z'\sin\theta)^2}{a^2} + \frac{y'^2}{b^2}, \\
x'\sin\theta + z'\cos\theta &= \frac{x'^2\cos^2\theta + z'^2\sin^2\theta - 2x'z'\cos\theta\sin\theta}{a^2} + \frac{y'^2}{b^2}, \\
x'\sin\theta + z'\cos\theta &= \frac{\cos^2\theta}{a^2}x'^2 + \frac{\sin^2\theta}{a^2}z'^2 - \frac{2\cos\theta\sin\theta}{a^2}x'z' + \frac{y'^2}{b^2}, \\
0 &= \frac{\cos^2\theta}{a^2}x'^2 + \frac{1}{b^2}y'^2 + \frac{\sin^2\theta}{a^2}z'^2 - \frac{2\cos\theta\sin\theta}{a^2}x'z' - x'\sin\theta - z'\cos\theta \qquad (17)
\end{aligned}$$

Comparing this result with the equation of `GQ` in Table 5.68 and specifying the corresponding parameters defines a paraboloid surface rotated by an angle θ around y axis. In this example, $\theta = 30°$ is given as the constant `c1`, while `c2` and `c3` are defined as $\sin\theta$ and $\cos\theta$, respectively.

Example 19: Example of Torus Surface

```
[surface]
  1  TZ    0 0 0  10 3 5
```

A torus surface, as shown in the center of Fig. 5.21, can be obtained by rotating an ellipse around a rotational axis outside of the ellipse. In PHITS, torus surfaces with rotational axes corresponding to the x, y, z axes can be specifying the symbols `TX,TY,TZ`, respectively. The equation of a torus surface with z axis as the rotational axis is given as

$$\frac{\left(\sqrt{x^2+y^2}-R\right)^2}{a^2}+\frac{z^2}{b^2}=1. \tag{18}$$

In this case, the ellipse has major and minor radii z and b, respectively, and a distance between the center of the ellipse and the rotational axis given by R. The center of the torus surface in Example 19 is set at the origin by setting the first, second, and third parameters of `TZ` to zero. The fourth, fifth, and sixth parameters are $A = R = 10$ cm, $B = b = 3$ cm, and $C = a = 5$ cm, respectively. The intersection line of the torus with the plane $z = 0$ is shown in the left panel of Fig. 5.21. The distance between the z axis and the center of the ellipse is 10 cm and the width of the ring is $5 \cdot 2 = 10$ cm. The intersection of the torus with the plane $y = 0$ is shown in the right panel of Fig. 5.21: note that only the region $x > 0$ is shown. The major and minor radii of the ellipse are 5 and 3 cm, respectively.

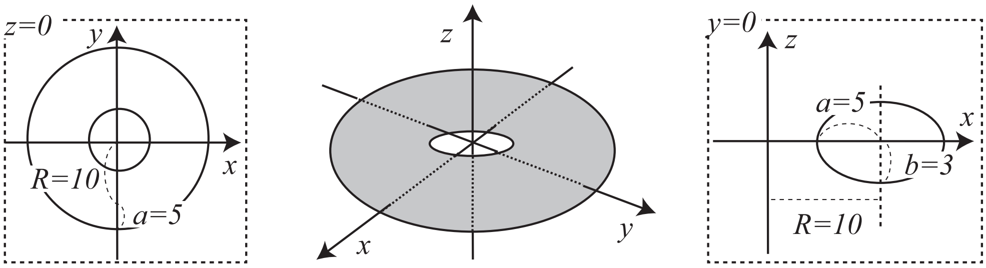


Figure 5.21: The torus surface defined in Example 19.

Example 20: Example of `BOX`

```
[surface]
  1  BOX   5 5 0  9 0 0  0 6 0  0 0 3
```

The symbol `BOX` can be specified to define the surfaces of an arbitrary rectangular solid. As parameters of `BOX`, the coordinates of the base point $P(x_0, y_0, z_0)$ and each component of three vectors $\boldsymbol{A}(A_x, A_y, A_z)$, $\boldsymbol{B}(B_x, B_y, B_z)$, and $\boldsymbol{C}(C_x, C_y, C_z)$ must be defined. Figure 5.22 shows the relation between the basepoint and the three vectors. In Example 20, the coordinate of the basepoint is $(5, 5, 0)$ and the three vectors are given as $\boldsymbol{A} = (9, 0, 0)$, $\boldsymbol{B} = (0, 6, 0)$, and $\boldsymbol{C} = (0, 0, 3)$. In using `BOX`, vectors $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}$ do not have to be parallel to the x, y, z axes, but they should be vertical to each other.

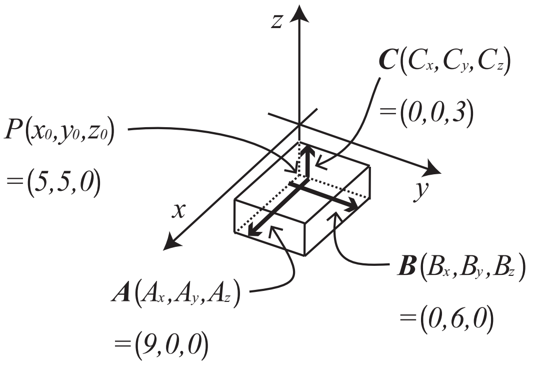


Figure 5.22: The rectangular surface defined in Example 20.

Example 21: Example of RPP

```
[surface]
  1   RPP    5 14  5 11  0 3
```

To define the surfaces of a rectangular solid surrounded by three sets of two planes parallel to the xy, yz, zx planes, the symbol RPP can be used. Unlike BOX, an arbitrary rectangular solid with surfaces not parallel to the x, y, z axes CANNOT be defined. The six parameters of RPP must be set by the user as the maximum and minimum values in the x, y, z coordinate system. In Example 21, $x_{\mathrm{min}} = 5, x_{\mathrm{max}} = 14$, $y_{\mathrm{min}} = 5, y_{\mathrm{max}} = 11$, and $z_{\mathrm{min}} = 0, z_{\mathrm{max}} = 3$ are specified as the six parameters. The result is the same rectangular solid defined in Example 20 and is shown in Fig. 5.23.

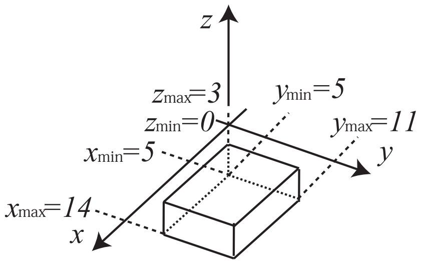


Figure 5.23: The rectangular surface defined in Example 21.

Example 22: Example of SPH

```
[surface]
  1   SPH    5 5 5  5
```

Symbol SPH can be used to define a sphere surface with an arbitrary coordinate as its center. The first, second, and third parameters are components of the coordinate (x_0, y_0, z_0) and the fourth parameter is the radius of the sphere. A sphere surface centered at $(5, 5, 5)$ with a radius of 5 cm is defined in Example 22. There is no functional difference between the symbols SPH and S.

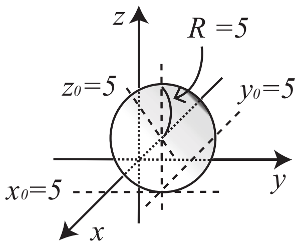


Figure 5.24: The sphere surface defined in Example 22.

Example 23: Example of RCC

```
[surface]
  1   RCC    5 5 0  0 0 10  5
```

The symbol RCC can be used to define an arbitrary cylinder surface. The parameters of RCC are the coordinates of the center of the bottom of the cylinder, $P(x_0, y_0, z_0)$, a vector from the bottom to the top, $\boldsymbol{H}(H_x, H_y, H_z)$, and the radius of the cylinder, R. The relation between these parameters is shown in Fig. 5.25. In Example 23, the coordinate $P(5, 5, 0)$ is the center of the bottom circle, the vector is $\boldsymbol{H} = (0, 0, 10)$, and the radius is 5 cm. Unlike symbols such as CX and C/X, it is useful to define a cylinder surface with a central axis NOT parallel to x, y, z axes.

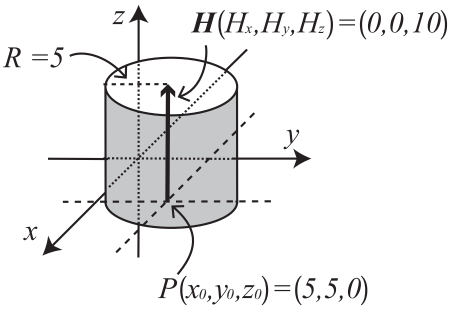


Figure 5.25: The cylinder surface defined in Example 23.

Example 24: Example of RHP

```
1:   [surface]
2:     1   RHP    0 0 0  0 0 10  5 0 0  2 -5 0  -2 -5 0
```

The symbols RHP or HEX can be used to define hexagonal prism surfaces as shown in the right panel of Fig. 5.26. As parameters, the center coordinate of the bottom of the prism, $P(x_0, y_0, z_0)$, a vector from the bottom to the top, $\boldsymbol{H}(H_x, H_y, H_z)$, and three vectors, $\boldsymbol{A}(A_x, A_y, A_z)$, $\boldsymbol{B}(B_x, B_y, B_z)$, and $\boldsymbol{C}(C_x, C_y, C_z)$ must be specified. Vectors $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}$ are required to define the hexagon comprising the bottom and top surfaces. The left panel of Fig. 5.26 shows the relation between point P and the three vectors; namely, the lengths and directions of the perpendicular lines between P and the three contiguous sides of the hexagon determine the three vectors. In Example 24, the origin is at the center of the bottom hexagon and the height along the z axis of the prism is 10 cm. The components of the origin $(0, 0, 0)$ are specified as the first, second, and third parameters of RHP, respectively, and those of the vector $\boldsymbol{H}(H_x, H_y, H_z) = (0, 0, 10)$ are given as the fourth, fifth, and sixth parameters, respectively. The seventh to fifteenth parameters are the x, y, z components of $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}$, respectively. The defined hexagon shown in the left panel of the figure is symmetric with respect to the y axis.

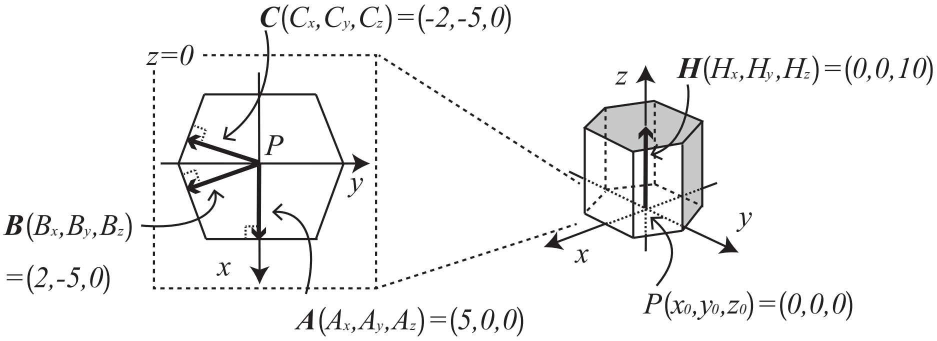


Figure 5.26: The hexagonal prism surface defined in Example 24.

Example 25: Example of REC

```
1:   [surface]
2:     1   REC    0 0 0  0 0 10  5 0 0  0 2 0
```

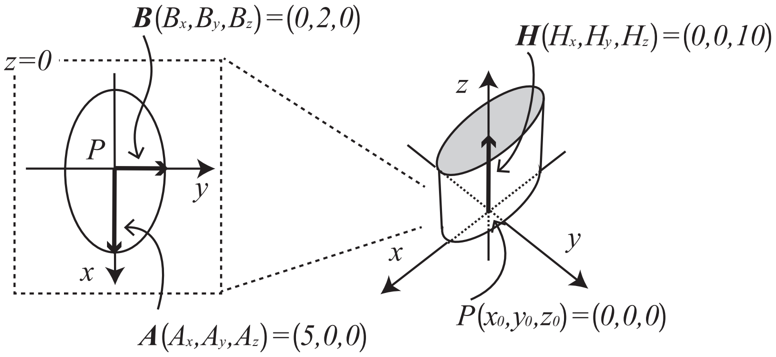


Figure 5.27: The elliptical cylinder surface defined in Example 25.

The symbol REC defines elliptical cylinder surfaces as shown in the right panel of Fig. 5.27. The parameters of REC are the coordinates of the center coordinate of the bottom surface, $P(x_0, y_0, z_0)$, the components of a vector from the bottom to the top, $\boldsymbol{H}(H_x, H_y, H_z)$, and the components of two vectors, $\boldsymbol{A}(A_x, A_y, A_z)$ and $\boldsymbol{B}(B_x, B_y, B_z)$. The relation between the point P and the two vectors $\boldsymbol{A}$ and $\boldsymbol{B}$ is shown in the left panel of Fig. 5.27. The major and minor axis vectors $\boldsymbol{A}$ and $\boldsymbol{B}$ must both be defined using initial point, P, which is the center of the ellipse. The surfaces of an elliptical cylinder with a height along the z axis of 10 cm are defined in Example 25. The center of the bottom is the origin; the first, second, and third parameters of REC are the coordinates of the origin $(0, 0, 0)$, and the fourth, fifth, and sixth parameters are those of vector $\boldsymbol{H}(H_x, H_y, H_z) = (0, 0, 10)$. The x, y, z components of $\boldsymbol{A}$ and $\boldsymbol{B}$ are given in the seventh to twelfth parameters. The vectors $\boldsymbol{A}$ and $\boldsymbol{B}$ are used to form the ellipse shown in the left panel of the figure.

Example 26: Example of TRC

```
[surface]
  1   TRC    0 0 0  0 0 10  5  2
```

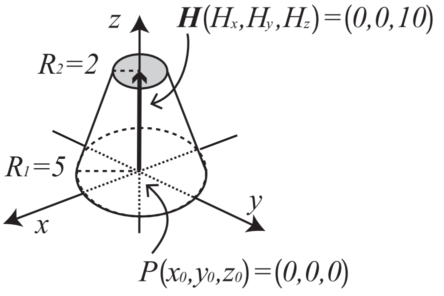


Figure 5.28: The cut cone surface defined in Example 26.

The surfaces of a truncated right-angle cone, as shown in Fig. 5.28, can be defined by specifying the symbol TRC. As parameters of TRC, each component of the center coordinate, $P(x_0, y_0, z_0)$, of the cone bottom, the components of a vector from the bottom to the top, $\boldsymbol{H}(H_x, H_y, H_z)$, and two radii of the bottom and top circles, R_1 and R_2, must be specified. In Example 26, the surfaces of the truncated right-angle cone of height 10 cm along the z axis are defined. The center of the bottom circle is the origin. The coordinates of the origin $(0, 0, 0)$ are specified as the first, second, and third parameters, respectively, and those of vector $\boldsymbol{H}(H_x, H_y, H_z) = (0, 0, 10)$ are given as the fourth through sixth parameters. The radius of the bottom is $R_1 = 5$ cm and that of the top is $R_2 = 2$ cm with R_1 and R_2 specified as the seventh and eighth parameters, respectively.

R_2 cannot be set to zero when defining TRC. To set a typical cone, i.e., a non-truncated cone, using TRC, R_2 should be set to a small value.

Example 27: Example of ELL (1)

```
[surface]
  1   ELL    3 3 0  -3 -3 0  9
```

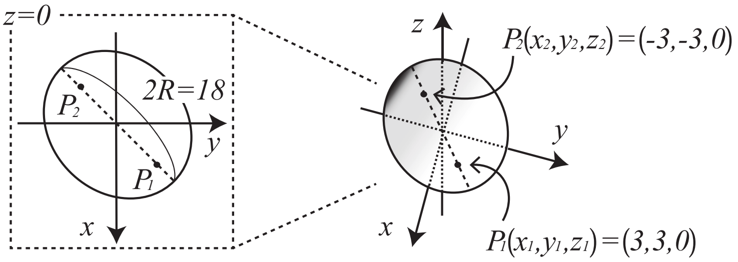


Figure 5.29: The spheroid surface defined in Example 27.

The surface of an ellipsoid of revolution (a spheroid) can be set by defining the symbol ELL. After defining an ellipse by specifying the coordinates of two focal points or the coordinate of its center, the ellipsoid of revolution is defined by rotating the ellipse around its major axis. Note that the meanings of the first six parameters of ELL depend on the sign of the seventh parameter, R. Example 27 is an example with $R > 0$. In this case, the coordinates of the two focal points and the radius of the major axis must be specified. The coordinate of the first focus, $P_1(x_1, y_1, z_1)$, is given by the first, second, and third parameters, while that of the second, $P_2(x_2, y_2, z_2)$, is given by the fourth, fifth, and sixth parameters. The seventh parameter is the radius of the major axis, R. The surface of the ellipsoid of revolution defined in Example 27 is shown on the right side of Fig. 5.29. The first and second focal points are $P_1(x_1, y_1, z_1) = (3, 3, 0)$ and $P_2(x_2, y_2, z_2) = (-3, -3, 0)$, respectively. The ellipse is on the xy plane, as shown in the left panel of the figure, with an angle between the major axis and the x axis of 45°. The length of the major axis is $2R = 18$ cm. The ellipsoid is defined by rotating this ellipse around the major axis.

Although an ellipsoid surface can also be defined using SQ, it is easier to define an ellipsoid with an inclined rotational axis using ELL. Note that only surfaces of ellipsoids of revolution can be defined using ELL.

Example 28: Example of ELL (2)

```
[surface]
  1  ELL   0 0 0  6 6 0  -6
```

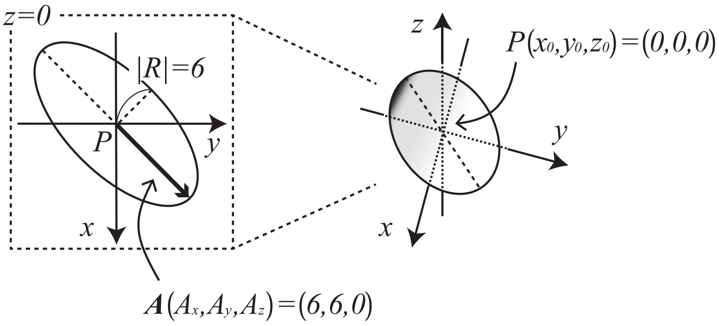


Figure 5.30: The spheroid surface defined in Example 28.

Example 28 apples to the case $R < 0$ in ELL. In this case, the center coordinate, P, major axis, $\boldsymbol{A}$, and radius of the minor axis of the ellipse must be defined. In this example, the center of the ellipse corresponds to the origin. The first, second, and third parameters of ELL are given as $P(x_0, y_0, z_0) = (0, 0, 0)$, while the fourth, fifth, and sixth parameters are $\boldsymbol{A}(A_x, A_y, A_z) = (6, 6, 0)$. The ellipse defined by these parameters is shown in the left panel of Fig. 5.30. The absolute value of the seventh parameter, R, which corresponds to the radius of the minor axis, is 6 cm. The ellipsoid surface obtained by rotating this ellipse around the major axis is shown in the right side of Fig. 5.30.

Example 29: Example of WED

```
[surface]
  1  WED   0 0 0  10 0 0  0 10 0  0 0 5
```

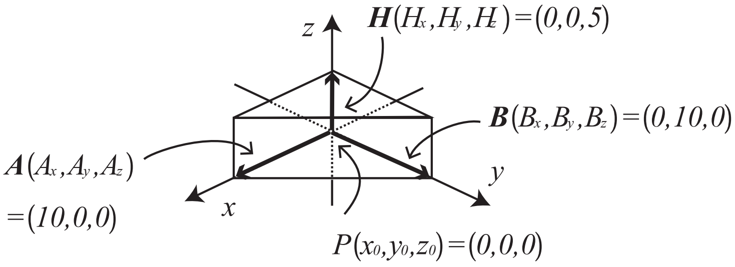

z
$\boldsymbol{H}(H_x,H_y,H_z)=(0,0,5)$
$\boldsymbol{B}(B_x,B_y,B_z)=(0,10,0)$
y
$\boldsymbol{A}(A_x,A_y,A_z)$
$=(10,0,0)$
x
$P(x_0,y_0,z_0)=(0,0,0)$

Figure 5.31: The wedge surface defined in Example 29.

A wedge surfaces, as shown in Fig. 5.31, can be obtained by specifying the symbol WED. Note that only a right triangle can be used as the bottom of a wedge defined by WED. In Example 29, one vertex of the bottom triangle $P(x_0, y_0, z_0) = (0, 0, 0)$ is given in the first, second, and third parameters of WED, respectively. Two vectors $\boldsymbol{A}(A_x, A_y, A_z) = (10, 0, 0)$ and $\boldsymbol{B}(B_x, B_y, B_z) = (0, 10, 0)$ with initial point P are specified in the fourth to ninth parameters. The triangle defined by the two vectors is a right triangle. The height vector of the wedge, $\boldsymbol{H}(H_x, H_y, H_z) = (0, 0, 5)$ is given in the tenth through twelfth parameters.

5.5.3 Surface definition by macro body

In using a surface defined by a macro body in the cell definition, '-' refers to the inside of the macro body and '+' is the outside. Each surface comprising a macro body can be used in the cell definition, in which case the macro body surface number should be written as '.' plus the macro body surface number.

The surface numbers are listed in Table 5.70.

Table 5.70: Macro Body Surface Numbers

Symbol	Macro Body Surface Number	Explanation
BOX	1	Surface vertical with the end of $(A_x,\ A_y,\ A_z)$.
	2	Surface vertical with the origin of $(A_x,\ A_y,\ A_z)$.
	3	Surface vertical with the end of $(B_x,\ B_y,\ B_z)$.
	4	Surface vertical with the origin of $(B_x,\ B_y,\ B_z)$.
	5	Surface vertical with the end of $(C_x,\ C_y,\ C_z)$.
	6	Surface vertical with the origin of $(C_x,\ C_y,\ C_z)$.
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		Only a user-defined surface number is used.
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 $(A_x,\ A_y,\ A_z)$.
	2	Opposite face for surface 1.
	3	Surface vertical with the end of $(B_x,\ B_y,\ B_z)$.
	4	Opposite face for surface 3.
	5	Surface vertical with the end of $(C_x,\ C_y,\ C_z)$.
	6	Opposite face for surface 5.
	7	Surface vertical with the end of $(H_x,\ H_y,\ H_z)$.
	8	Surface vertical with the origin of $(H_x,\ H_y,\ H_z)$.
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		Only a user-defined surface number is used.
WED	1	Surface including the top and bottom hypotenuses.
	2	Surface including $\boldsymbol{B}$ and $\boldsymbol{H}$.
	3	Surface including $\boldsymbol{A}$ and $\boldsymbol{H}$.
	4	Surface including $\boldsymbol{A}$, $\boldsymbol{B}$, and the end of $\boldsymbol{H}$.
	5	Surface including $\boldsymbol{A}$, $\boldsymbol{B}$, and the origin of $\boldsymbol{H}$.

5.6 [Cell] section

5.6.1 Formats

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

Only C and $ can be used as comment marks in this section; in particular, # cannot be used as a comment mark as this character is used for the cell definition. To use continuation lines, it suffices to place at least five blanks at the line head instead of using the line sequential mark at the end of the line.

The [cell] section is defined in this data order: cell number, material number, material density, cell definition, and cell parameter as keyword style. These are explained in Table 5.71. The format is shown below.

```
[ Cell ]
cell number     mat.  number     mat.  density     cell def.      cell parameter
```

In the LIKE *n* BUT format, the cell parameter format and repeated structures with lattices can be used: see Sec. 5.6.5 for some examples showing how to use this format. The cell parameters are listed and explained in Table 5.72.

Table 5.71: Cell definition format

item	explanation
cell number	Any number from 1 to 999,999 can be used.
material number	Set 0 for void, −1 for outer void, or use a material number defined in the [material] section.
material density	If the cell is void or outer void, no input. When the given value is positive or negative, it is a particle density [10^{24} atoms/cm^3] or mass density [g/cm^3], respectively. A composition ratio defined in the [material] section is used. Thus, different density materials with the same composition as the original can be set in this section. A new parameter, matadd, is used to add different material numbers.
cell definition	Cell geometry is defined by both surface numbers in the [surface] section and the Boolean operators, ␣(blank)(AND), :(OR), and #(NOT). Parentheses (and) can be also used. See Sec. 5.6.2 for details.
LIKE *n* BUT	A cell using this format is the same as an *n* cell except for using only parameters described after BUT.
cell parameter	The format is keyword=value. As a keyword, VOL(volume), TMP(temperature), TRCL(transform), U(universe), LAT(lattice), or FILL can be used. In the LIKE *n* BUT format, MAT(material) and RHO(density) can be used as well.

When defining cells with the same material number but different densities, the cells assume material numbers that differ from that of the first cell.

In operation of the cell definition, ␣(blank) has a higher priority than :.

Table 5.72: Cell parameters

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 performed using the defined coordinate transform number defined in the `[transform]` section or the transform format.
U	Universe number; using this, the number of the universe including the cell is defined. Any number value can be used from 1 to 999,999. See Sec. 5.6.3 for details.
LAT	Lattice number. Setting `LAT=1 or 2` defines a quadratic or hexangular prism, respectively: see Sec. 5.6.4 for details.
FILL	Set universe numbers to fill the cell with the universe.
MAT	This is used with the `LIKE` *n* `BUT MAT=`*m* format. Using `MAT`, a cell can be duplicated except with its material number changed to *m*.
RHO	This is used with the `LIKE` *n* `BUT RHO=`*x* format. Using `RHO`, a cell can be duplicated except with its density changed to *x*.

5.6.2 Description of cell definition

Cells are defined by treating regions divided by surfaces defined in the `[surface]` section. When giving a cell definition, in some cases the concept of ‘surface sense’ must be used to create a distinction between two regions divided by the surface corresponding to an equation, $f(x, y, z) = 0$; in other cases, the Boolean operators, $\sqcup$(blank)(AND), `:`(OR), and `#`(NOT) must be used to treat the regions.

‘Surface sense’ defines one region including a point (x_0, y_0, z_0) by treating $f(x_0, y_0, z_0) > 0$ in a ‘positive sense’ and the region $f(x_0, y_0, z_0) < 0$ in a ‘negative sense.’ To define a region in the positive sense, a positive surface number is written in the cell definition space; conversely, a negative surface number is used in the definition to denote a negative sense. An example of the definition of sense is shown below.

Example 30: `[cell]` section example (1)

```
[ Cell ]
  1   0  -10
  2  -1   10
[ Surface ]
  10  SZ  3  5
```

The tenth surface represents a sphere with a radius of 5 cm. Because the inside of this sphere is in the negative sense, the first cell is defined using a negative number `-10`. The outer void is explicitly defined as the second cell. This example produces the virtual space shown in Fig. 5.32.

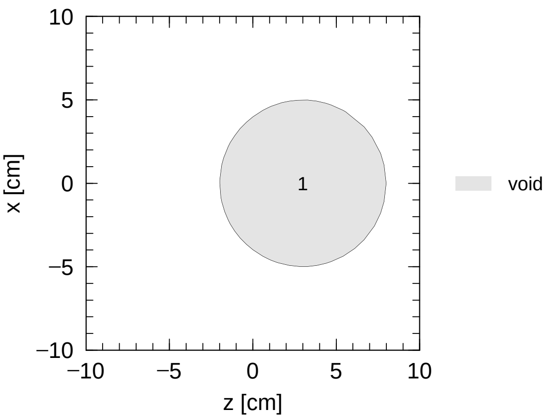


Figure 5.32: Result of `[cell]` section example (1).

In some cases, the treatment of a region in the cell definition involves the use of Boolean operators. The symbols ⊔(blank), `:`, and `#` denote the intersection (AND), union (OR), and complement (NOT), operators, respectively. Parentheses, e.g., `(` and `)`, can be used to combine some regions. The second example in this section uses ⊔(blank) and `#`.

Example 31: `[cell]` section example (2)

```
 1:   [ Cell ]
 2:     1   0  11  -12  13  -14  15  -16
 3:     2  -1  #1
 4:   [ Surface ]
 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 second line, the three numbers without minus signs correspond to the positive sense regions of the eleventh, thirteenth, an fifteenth surfaces, while those with minus signs correspond to the negative sense regions of the twelfth, fourteenth, and sixteenth surfaces. A region surrounded by these surfaces is defined with ⊔(blank) as the first cell, which is the interior of a 12-cm cube. The outside of the cube is defined by the complement operator `#` as the outer void. Figure 5.33 shows the result of this example.

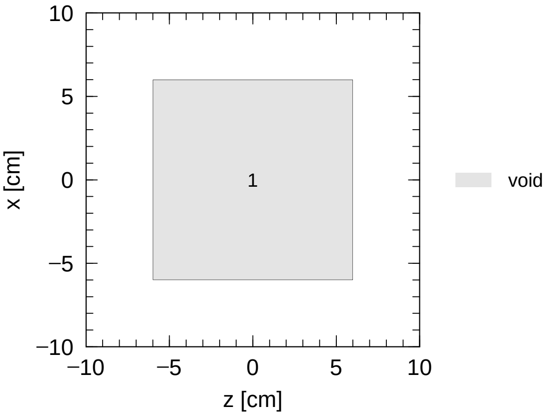


Figure 5.33: Result of `[cell]` section example (2).

The next example uses `:` and parentheses to combine the sphere in the first example and the cube in the second example.

Example 32: `[cell]` section example (3)

```
 1:   [ Cell ]
 2:     1   0  -10 : (11  -12  13  -14  15  -16)
 3:     2  -1  #1
 4:   [ Surface ]
 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
```

The numbers surrounded by the parentheses in the second line correspond to the region of the first cell in example (2). In this example, a region combining the inside of the cube with the inside of the sphere in example (1) is defined using the union operator `:` as the first cell. The result is shown in Fig. 5.34.

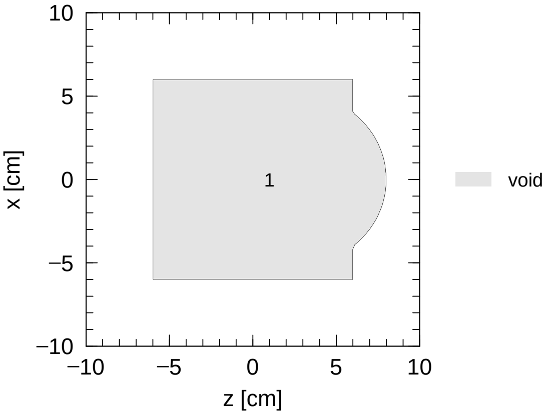


Figure 5.34: Result of `[cell]` section example (3).

The next example shows the division of a cube into two regions by a spherical surface.

Example 33: `[cell]` section example (4)

```
 1:   [ Material ]
 2:     mat[1]  1H 2  16O 1
 3:   [ Cell ]
 4:     1   0       -10
 5:     2   1  1.0   10 (11  -12  13  -14  15  -16)
 6:     3  -1        #1 #2
 7:   [ Surface ]
 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 as in example (3). In the fifth line, the second cell is defined with ␣(blank) as an overlap region between the outside of the sphere, which is the tenth surface, and the inside of the cube defined by the parentheses. The cell is filled with water as defined in the `[material]` section and the result is shown in Fig. 5.35. The interior of the sphere is the first cell, which is filled with void.

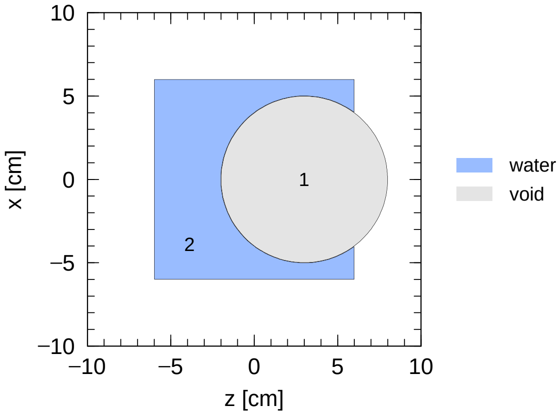


Figure 5.35: Result of `[cell]` section example (4). The first and second cells are filled with void and water, respectively.

5.6.3 Universe frame

In PHITS, the cell parameter U can be used to define a region of the main space used for particle transport calculation that can be translated to a corresponding region in any universe. This function is very useful for setting the repeated structures introduced in Sec. 5.6.5.

An example using three spaces (one main space and two universes) and shown in Fig. 5.36 is explained below. The main space comprises two rectangular solids. One universe includes a cylinder filled with water, while the other has an iron cylinder surrounded by water. The first cell is filled with a region of universe 1, while the second cell is filled with a region of universe 2.

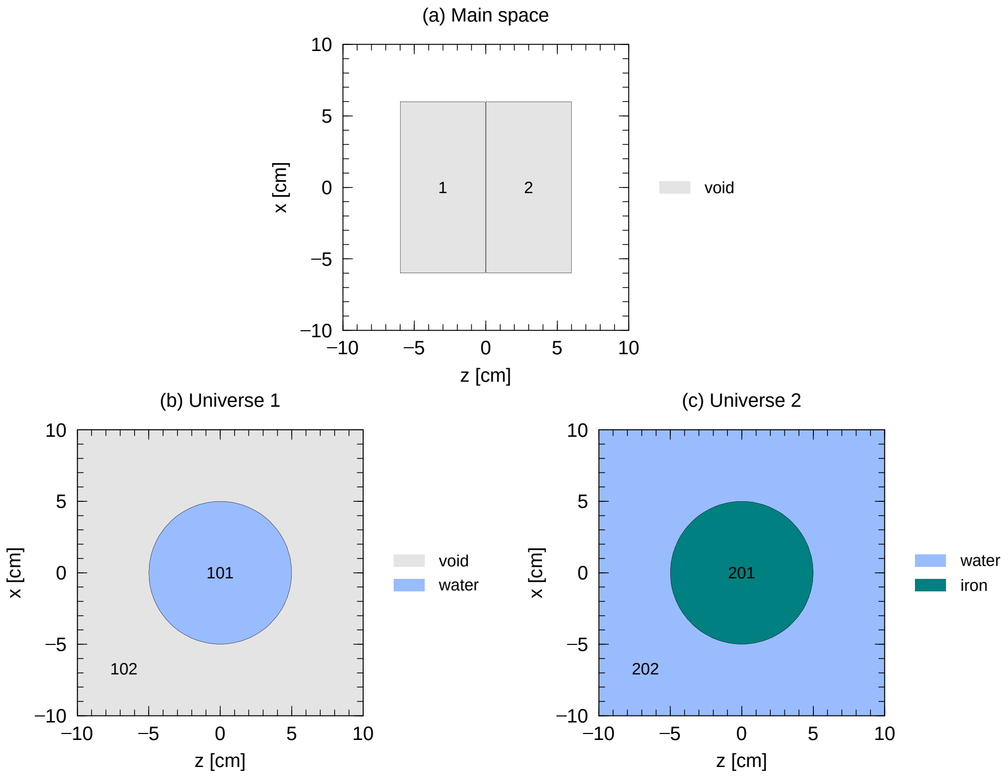


Figure 5.36: (a) Two rectangular solids. (b) Cylinder filled with water. (c) Iron cylinder in water.

Example 34: [cell] section example (5)

```
[ Material ]
  mat[1]  1H 2  16O 1
  mat[2]  Fe 1
[ Cell ]
    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
[ Surface ]
  10  CY   5
  11  PX  -6
  12  PX   6
  13  PY  -6
  14  PY   6
  15  PZ  -6
  16  PZ   6
  17  PZ   0
```

Universes `1 and 2` are defined in the seventh and eighth lines and in the ninth and tenth lines, respectively, using cell parameter `U`. These universes have similar structures in which a cylinder is placed at the origin of the coordinate space, but their components inside and outside the cylinder differ, as shown in Fig. 5.36. In the fifth and sixth lines, the first and second cells are defined respectively as regions filled with the corresponding part of each universe using the cell parameter `FILL`. The result of this example is shown in Fig. 5.37, in which it can be seen that the first cell comprises the 101st and 102nd cells in universe 1, while the second cell comprises the 201st and 202nd cells in universe 2.

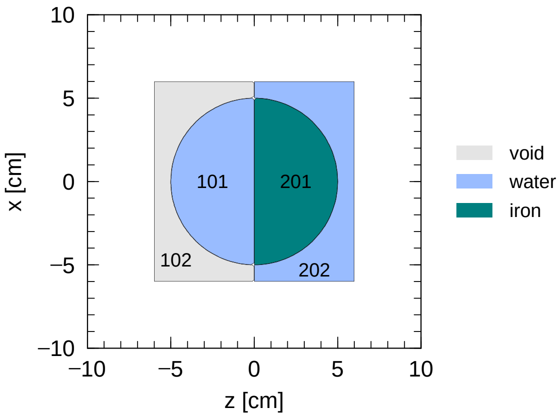


Figure 5.37: Result of `[cell]` section example (5).

It is not possible to use an undefined region from one of the universes. If the 102nd cell is not defined in the eighth line as a void region, the first cell cannot be filled with universe 1. Not also that all universes have the same coordinate system definition, with the position of the origin, directions of x, y, and z-axes, and scale of the space in any universe agreeing with those in any other universe. If a different value of `PX` is in the fourteenth and fifteenth lines, the cube will not include some of the cylinder, as shown in Fig. 5.38.

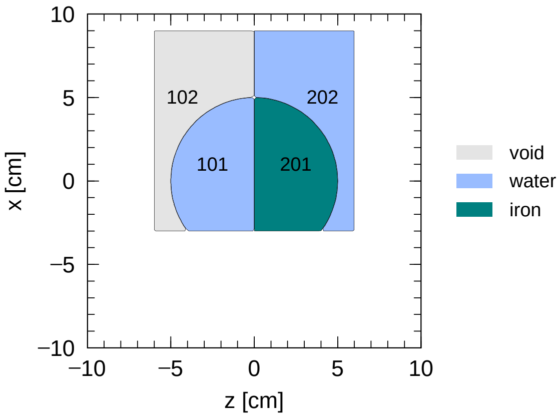


Figure 5.38: Result of `[cell]` section example (5) with the region is shifted in the x-direction.

5.6.4 Lattice definition

The cell parameter `LAT` (lattice parameter) is very useful for making repeated structures. In this section, the definition of a unit lattice structure and its simple use are explained using some examples: see Sec. 5.6.5 for a more practical description.

The quadratic and hexangular prisms shown in Fig. 5.39 can be used as a unit structure by setting `LAT=1` or `LAT=2`, respectively. A universe can be constructed using repeated structure of such a lattice, and any region can be filled using this universe. Note that each unit must also be filled with another universe defined with any material or void. The numbering of each unit component in Fig. 5.39 corresponds to the surface number order written in the cell definition; the lattice coordinate system, which will be explained below, depends on this order.

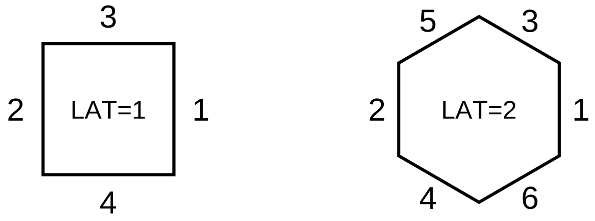


Figure 5.39: Unit structure of lattice.

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

Example 35: `[cell]` section example (6)

```
 1:   [ Material ]
 2:     mat[1]  1H 2  16O 1
 3:   [ Cell ]
 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:   [ Surface ]
 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 fifth line, a unit cell with `LAT=1` is defined using four surface numbers. Setting `U=1` defines as the repeated structures of this unit, which is filled with universe 2 defined in the sixth line. Because the cross section of the unit in the *x*-*z* plane is a square four cm per side, the first cell defined in the fourth line as a 12 cm cube has nine blocks, as shown in Fig. 5.40. Note that the unit has an infinite length in the *y* direction of universe 1 because of only four surfaces are defined. To define a finite-length prism, `-24 23` must be added to the cell definition in the fifth line.

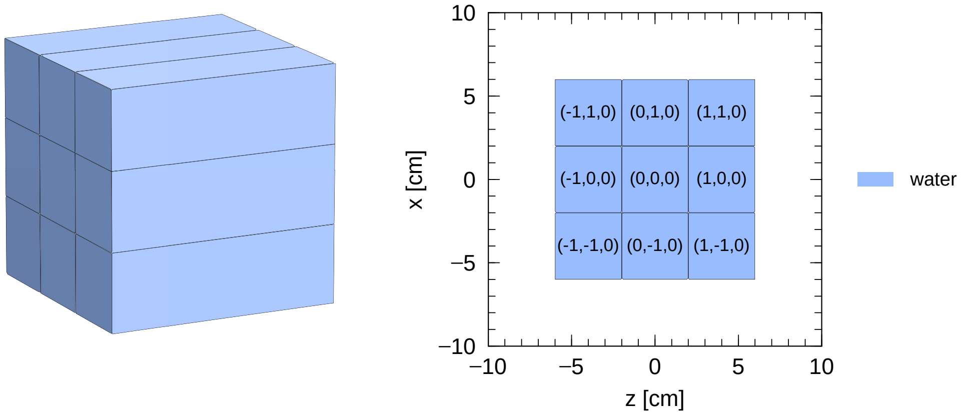


Figure 5.40: Result of `[cell]` section example (6) in 3D (left) and 2D (right) images.

To distinguish cells in the repeated structure, each cell is placed at the lattice coordinate (s,t,u), as shown in the right panel of Fig. 5.40. Note that the ordering this coordinate notation corresponds to the general coordinate ordering (x, y, z) and are defined by the order of surface numbers written in the cell definition. To specify any cell using `mesh=reg` in a tally section, the lattice and universe styles `(201 < 101[-1 0 0] < 1)` can be used, where the lattice coordinate is represented by `[s t u]`: see Sec. 6.1.2 for more information on this format. The lattice coordinates can be viewed using the `[t-gshow]` tally with `output=7 or 8`.

The following example involves a hexangular prism (`LAT=2`).

Example 36: `[cell]` section example (7)

```
 1:   [ Material ]
 2:     mat[1]  1H 2  16O 1
 3:   [ Cell ]
 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:   [ Surface ]
 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 fifth line using the six surfaces defined in the seventeenth through twenty third lines. The hexagonal prism is restricted in the y-direction by `-24 23` in the cell definition and is filled with the universe 2, i.e., water, as specified in the sixth line. The first cell has the repeated structure defined in universe 1. Figure 5.41 shows the result of this example. It can be seen that some prisms near the edges of the first cell, which is defined as a 12 cm cube, are only partly used. The directions of the lattice coordinate shown in the right panel depend on the order of the surface number written in the cell definition. To specify a cell using `mesh=reg` in a tally section, the lattice and universe styles `(201 < 101[-2 0 0] < 1)`, where the lattice coordinate is represented by `[s t u]`, can be used: see Sec. 6.1.2 for more information on this format. The lattice coordinates can be shown using the `[t-gshow]` tally with `output=7 or 8`.

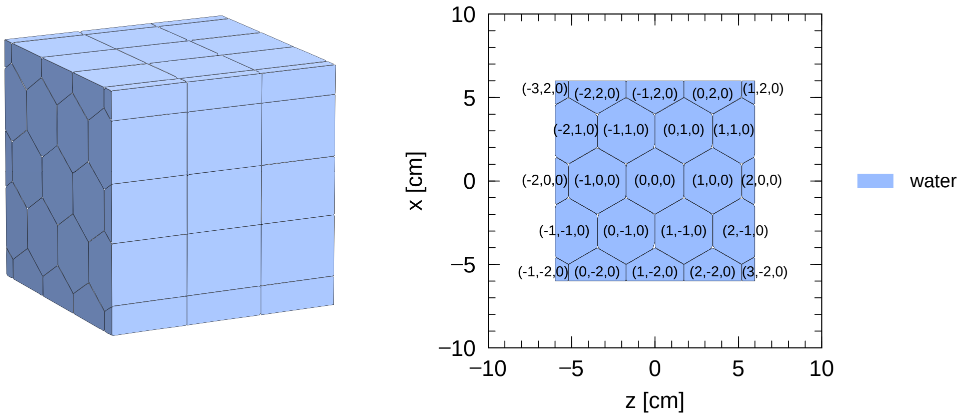


Figure 5.41: Result of `[cell]` section example (7) in 3D (left) and 2D (right) images.

5.6.5 Repeated structures

There are several simple procedures in PHITS that can be used to create repeated structures in which the same or similar units are repeated. Using the lattice parameter described in Sec. 5.6.4 is one such method; another is the `LIKE` *n* `BUT cell parameter` format.

`LIKE` *n* `BUT` cell parameter

Using this format, a cell that differs slightly from an original cell can be created. In this format, the cell parameters following the `BUT` statement differ from the those in the original cell (*n*). The cell parameters that can be used in this format are shown in Table 5.72. In the following example, two cell parameters, `TRCL` and `MAT`, are used.

Example 37: `[cell]` section example (8)

```
 1:   [ Material ]
 2:     mat[1]  1H 2  16O 1
 3:     mat[2]  Fe 1
 4:   [ Cell ]
 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:   [ Surface ]
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:   [ Transform ]
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 and placed at the origin of the coordinate system is defined in the sixth line. The interior of this cube is the second cell, which is regarded as the original cell in this example. In the seventh and eighth lines, respectively, the third and fourth cells are defined using the `LIKE` *n* `BUT` format with $n = 2$. Figure 5.42 shows the result of this example. The coordinate system of the third cell is transformed using the cell parameter `TRCL=1`, where the coordinate transform number 1 is defined in the nineteenth line of the `[transform]` section. The coordinate system of the fourth cell is transformed using `TRCL=2` and the interior material of the cell is replaced with iron defined using the material number 2 in the third line.

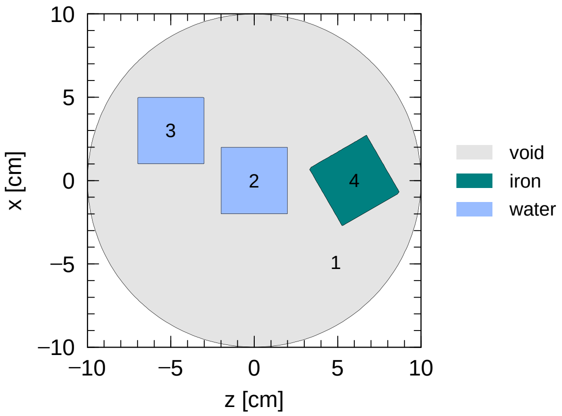


Figure 5.42: Result of `[cell]` section example (8).

Nesting structure with lattice

A nesting structure can be used based on the universe frame described in Sec. 5.6.3.. For example, universe 1 can be filled with universe 2, and universe 2 can be filled with universe 3, with this process continued following a user-defined nesting structure. The maximum number of nestings is 10, which corresponds to the parameter `mxlv` given in the file “param.inc.”

In the next example, there are nine square poles defined using `LAT=1`; three of these have unique structures.

Example 38: `[cell]` section example (9)

```
 1:   [ Material ]
 2:     mat[1]  1H 2  16O 1
 3:     mat[2]  Fe 1
 4:   [ Cell ]
 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:   [ Surface ]
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
```

The definition of the first cell in the fifth line and the lattice unit in the sixth line are the same as in the `[cell]` section example (6). However, the format of the cell parameter `FILL` in the seventh and eighth differs: in the seventh 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) following the order $(-1, -1, 0), (0, -1, 0), (1, -1, 0), (-1, 0, 0), \ldots, (1, 1, 0)$; in other words, a lattice at $(-1, -1, 0)$ is filled with universe 2 and one at $(1, -1, 0)$ is filled with universe 3. Universe 2 is defined in the ninth line as a space filled with water, while the universe 3 is defined in the tenth and eleventh lines has an iron cylinder centered at the origin. The result of this example is shown in Fig. 5.43, in which it can be seen that three lattices at $(1, -1, 0), (0, 0, 0)$, and $(-1, 1, 0)$ have an iron cylinder. Specifying a cell using `mesh=reg` in the tally sections can be done using the lattice and universe styles as `(302 < 101[0 0 0] < 1)`, where the lattice coordinate is represented by `[s t u]`: see Sec. 6.1.2 for more information on this format.

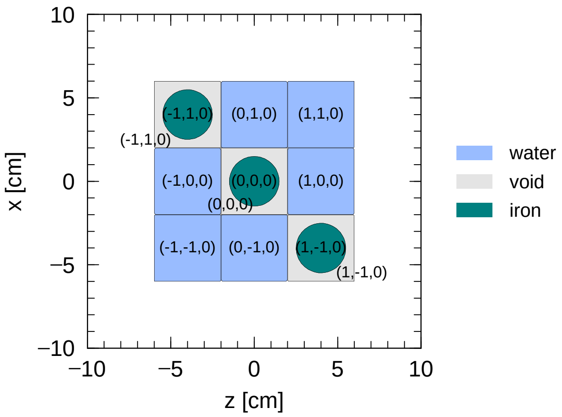


Figure 5.43: Result of `[cell]` section example (9).

A more complex example is shown below.

Example 39: `[cell]` section example (10)

```
 1:   [ Material ]
 2:     mat[1]  1H 2  16O 1
 3:     mat[2]  Fe 1
 4:   [ Cell ]
 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:   [ Surface ]
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 formed by this input is shown in Fig. 5.44, in which there are nine square poles defined using the lattice parameter. Three of the poles comprise four units of the other lattice. In the eighth line, `(1 0 1)` denotes a transformation of the coordinate system in which the origin is shifted by 1 cm in both the *x*- and *z*-directions. The lattice and universe styles `(402 < 301[-1 -1 0] < 101[0 0 0] < 1)`, where the lattice coordinate is represented as `[s t u]`, can be used to specify any cell using `mesh=reg` in the tally sections: see Sec. 6.1.2 for more information on this format.

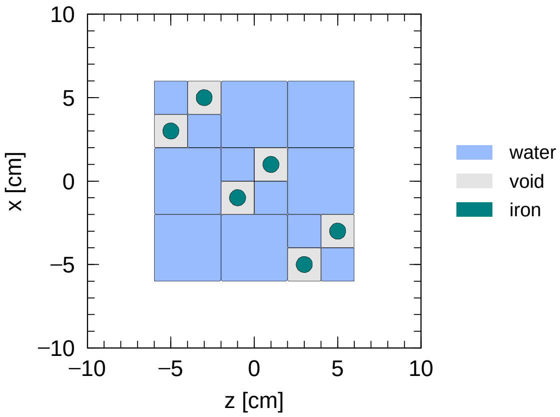


Figure 5.44: Result of `[cell]` section Example (10).

Voxel phantom

In PHITS, a virtual space can be created using voxel phantoms for calculation on complex structures such as the human body or an organism. To define a voxel phantom, a small cube must first be defined as the unit of a lattice with `LAT=1`. This unit is then repeated to define a large-size structure. Each unit can then be filled with a universe, which itself is filled with biological matter, e.g., compounds of carbon and water.

In the example below, a 10-cm cube comprising 125 (5 × 5 × 5) 2 cm cubes (voxels) is described.

Example 40: `[cell]` section example (11)

```
 1:  [ Material ]
 2:    mat[1]  1H 2   16O 1
 3:    mat[2]  Fe 1
 4:  [ Cell ]
 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:  [ Surface ]
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 the voxel unit, the 2 cm cube is defined in the twenty fourth line. The first cell, which is located within a 10-cm cube, has a repeated structure as defined in the fifth line. The region of the lattice coordinate space is determined in the seventh line. The order of voxels in the eighth through twelfth lines is as follows: $(-2, -2, -2), (-1, -2, -2), \dots, (2, 2, 2)$, which represent the lattice coordinates. In the eighth through twelfth lines, `2` means universe 2, which is void, while `3` and `4` correspond to universes 3 and 4, which are of iron and water, respectively. Figure 5.45 shows the results of this example —a distorted iron box with regions of water within. To specify any cell using `mesh=reg` in tally sections, the lattice and universe styles `(401 < 101[0 0 0] < 1)`, where the lattice coordinate is represented by `[s t u]`, can be used: see Sec. 6.1.2 for more information. Note that formats such as, e.g., `(301 < 101[-2:2 -2:2 -2:2] < 1)` cannot be used because not all `101[-2:2 -2:2 -2:2]` cells have a 301st cell.

Figure 5.45: Results of the [cell] section example (11) in 3D images. The structure in the right panel is equivalent to that in the left panel with part of its iron surface removed.

From the PHITS version 3.09, the array of universe numbers at the 8th through 12th lines of the example 40 can be expressed in a compressed format. For voxel phantoms, the universe array contains long ques of continuous same universe numbers and the compressed format is designed to reduce the size of this array. Thus the redution of the file size and computational time for read and write of the huge voxel data can be achieved. In the compressed format, the array of the continous universe numbers are given by the number of the continuous same universe with minus sign followed by the universe number. Following this rule, the array of universe number at the 8th through 12th lines of the example 40 can be rewritten as follows.

Example 41: `[cell]` section example (11')

```
 8:            2 -11 3 2 -17 3 -1 2 -2 3
 9:            4 3 2 -1 3 -1 2 -12 3 -2
10:            2 3 4 -2 3 2 3 -2 2 -12 3
11:            -1 2 -1 3 4 3 2 -2 3 -1 2
12:            -17 3 2 -11
```

In the example 40 after the universe number 2 are aligned 12 times, the universe number 3 comes. On the other hand in the example 41, the second number -11, which comes after the first number 2, specifies that the universe number 2 given before was replicated 11 times afterword.

To enable save computational time, `ivoxel` can be specified in the `[parameters]` section. Performing PHITS calculation with `ivoxel` set =2 causes voxel data to be output in binary to `file(18)` and then stops the calculation. From the next calculation with `ivoxel` set =1, data output is omitted, which reduces the calculation time. If a very large amount of voxel data is used, it may be more convenient to use `infl`.

Use of tetrahedron geometry

In this section, a definition and the basic usage of tetrahedron geometry are given using an example. Tetrahedron geometry is a type of polygon mesh geometry that can express complex geometries by combining tetrahedrons of varying sizes.

PHITS adopts two tetrahedron geometry formats. One is the format for the tetrahedral mesh generator, TetGen and the other is the format for the finite element analysis program NASTRAN.

The TetGen format is represented by two files:

- the node file (a list of the xyz space coordinates of the tetrahedron nodes);
- the element file (a list of four nodes IDs for each tetrahedron element).

In PHITS, the node and the element files have a common name with different suffixes—".node" and ".ele," respectively: please refer to the ppt file or sample input file in `\phits\utility\TetraGEOM\` for a more detailed explanation.

Examples of a node file and an element file are given below.

Example 42: Example of node file

```
1:   # Simple two elements
2:   5   3  0  0
3:   #  pointID x y z
4:   1    0.0  -2.5  -4.0
5:   2   -4.0  -2.5   0.0
6:   3    4.0  -2.5   0.0
7:   4    0.0  -2.5   4.0
8:   5    0.0   2.5   0.0
```

The node file begins with the number of nodes and the number of dimensions (second line). PHITS operates only in three-dimensional geometry, and the two last zeros in the second line are not used in PHITS. The following lines (below the third line) represent a list of nodes and their xyz space coordinates in the order

```
[node no.] [x] [y] [z]
```

Example 43: Example of element file

```
 1:   # Simple two elements
 2:   2  4  1
 3:   # elementID point(1:4) universe
 4:   1      1  2  3  5    1001
 5:   2      5  2  3  4    1002
```

The element file begins with the number of elements, the number of nodes comprising a tetrahedron (= 4 in PHITS), and the number of information points tagged to the element (=1 in PHITS) [second line]. The following lines (below the third line) give a list of 4-node IDs, each corresponding to a tetrahedron, in the order

```
[element no.] [node 1] [node 2] [node 3] [node4] [element universe no.]
```

The element no. represents the ID of the tetrahedron element. The node no. specifies the list of 4-node IDs in the node file. The element universe no. specifies the universe used to fill the tetrahedron element; the definition of this universe should be given in the PHITS input file (See Example 44). Arbitrary comment lines starting with # can be inserted into any line of the node or element file.

The node and element files can be created using the Tetrahedral Mesh Generator (TetGen) software, which can convert general polygon mesh data to tetrahedron geometry with some effort. TetGen can be obtained free of charge from the following source:

`http://wias-berlin.de/software/tetgen/`

For details on TetGen, please refer to the manual in the link above.

PHITS adopts another format, which is the bulk data format of NASTRAN. This format is a generally accepted in many software of structural analysis and computational fluid dynamics (CFD). PHITS uses the bulk data format defined using GRID and CTETRA and assumes those are written in a file with .bdf extention. The explantion of the buld data format is omitted here because the file will be deduced automatically from some software which is used to create the tetrahedron geometry. By adopting the same format for the tetrahedral geometry in PHITS as structural analysis or CFD, seamless coupled analysis between those studies and radiation transport calculation becomes possible. For the details, a document will be uploaded in the PHITS homepage

`http://phits.jaea.go.jp/`

which explains how to conduct such a coupled analysis.

The method for using tetrahedron geometry in a PHITS input file is explained below.

Example 44: [cell] section example (12)

```
 1:   [Material]
 2:   mat[1] 14N 78.1  16O 20.9  40Ar 0.93
 3:   mat[2] 1H 2  16O 1
 4:   mat[3] 56Fe 1
 5:   [Surface]
 6:   10  rpp   -5.0  5.0  -3.0  3.0  -5.0  5.0
 7:   20  rpp   -7.0  7.0  -5.0  5.0  -7.0  7.0
 8:   90  so     500.0
 9:   [Cell]
10:   101   1 -0.001205 -20  U=1  LAT=3  tfile=Tetra  TSFAC=1.0
11:     1   0           -10  FILL=1
12:     2  -1            10
13:   201   2 -1.0      -90  U=1001
14:   202   3 -7.874    -90  U=1002
```

Example 45: [cell] section example (13)

```
10:   101   1 -0.001205 -20  U=1  LAT=3  nfile=Tetra  TSFAC=1.0
```

To use tetrahedron geometry, a region must be defined as a rectangular shape (using the surface symbol RPP). This region should contain all of the nodes of the tetrahedrons that will be created but should not be so large that it incurs unnecessary computational costs. At the tenth line in the example code, a region with cell no. 101 is defined as a rectangular shape specified by surface number 20 as a 14 cm × 10 cm × 14 cm rectangular box. Setting the LAT option =3 declares the use of a tetrahedron geometry defined by the node and element files with names specified by

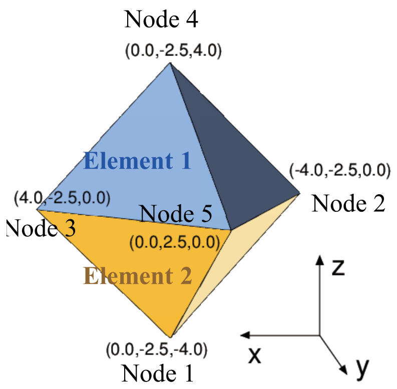


Figure 5.46: Results of the [cell] section example (12) in 3D images.

TFILE. For a file of the NASTRAN bulk data format, the name should be specified accompanied by file NFILE as in the example 45. The file of the name with the suffix .bdf will be imported. Note that the upper and lower cases of the file name differ between Mac and Linux. Using the TSFAC factor allows the size of the tetrahedron geometry to be scaled up or down by multiplying the coordinates of the nodes by the value of the factor. The material given in cell 101 is used to fill all of region 101 aside from the tetrahedron interiors. In the eleventh line the region with cell no. 1 is defined as a rectangular box of dimensions 10 cm × 6 cm × 10 cm. Using the FILL option, this region is filled by the tetrahedron geometry of cell no. 101. The Tetrahedron Geometry setting (LAT=3) should be used together with the universe and fill nest structures in the same manner as in lattice structure specification; the universes included in the element file should be defined in the same manner as in the lattice structure (this is done in lines 13 and 14 in this example, where element no. 1 is filled by material 2 (water), while element no. 2 is filled by material 3 (iron).

When the number of universes used in the tetrahedron geometry increases, the cost manually creating the cells done in lines 13 and 14 in the example 44 becomes heavy. By specifying the option itetauto=1 in the parameter section, the cells corresponding to the universes for tetrahedron geometry will be automatically created. With this option, extreamly large surface no. 5000 and cells from no. 5001 to no. 5000+n are added to those given in the PHITS input file and thus use of those no. should be avoided to use this option, where n is number of universes used in the tetrahedron geometry. Density of the added cells are automatically defined by reading PSOLID and MAT tabs when a file with the NASTRAN bulk data format is used. For a TetGen format file, density information should be provided by an external file. The file name needs to be a common but with .txt suffix. The universe no. and its density should be specified for each line as 46. The addional cells will be provied with the same material no. as the cell. Error messages as shown in 47 will be displayed by executing PHITS without specifying the materials corresponding to the universes in the tetrahedron geometry. The materials from 5001 to 5000+n should be defined according to these messages. This automatic process can be verified by cheking the input echo lines in the phits.out file, where the additional surface and cells will be inserted.

Example 46: [cell] Example of txt file

```
1001 -1.0
1002 -7.874
```

Example 47: [cell] Example of error message

```
*** ERROR : undefined material
 TETRA material (MID):  1001
   should be defined as material ID number:  5001
*** ERROR : undefined material
 TETRA material (MID):  1002
   should be defined as material ID number:  5002
```

5.7 [Transform] section

5.7.1 Formats

Coordinate transformation can be defined in this section. The transformation is available in `[source]`, `[surface]`, `[cell]` section, tallies with `r-z`, `xyz` mesh, and definition of magnetic and electro-magnetic fields. Only `C` and `$` can be used as a comment mark. Using `[transform]`, only rotation and translation of the coordinate can be set, and enlarge or shrink cannot be set.

Formats are shown below.

`[ Transform ]`

`TR`n O_1 O_2 O_3 B_1 B_2 B_3 B_4 B_5 B_6 B_7 B_8 B_9 M

The format with $*$, $*$TRn, is also used. When $*$TRn, components of a rotation matrix B_i can be given by angles in the unit of [degree].

Table 5.73: Transformation parameter

item	explanation
n	Transform number. Use any number from 1 to 999,999.
O_1, O_2, O_3	x, y, z components for translation.
$B_1 \sim B_9$	Components of rotation matrix. Please see Eqs. (19)~(22) in more detail. The determinant of the matrix is automatically adjusted to 1.
M	Option for equation of coordinate transformation. = 1: translation is performed after rotation. = -1: rotation is performed after translation. = ± 2: these options need only B_1, B_2, B_3 to define rotation angles around z, y, x axes, respectively. The order of the rotations is z, y, x. The unit of the angles is radian when TRn, and that is degree when $*$TRn (the format with $*$). The mathematical definition of the transformation of $M = 2$ and -2 is the same as $M = 1$ and -1, respectively.

Default values are shown below.

`TR`n 0 0 0 1 0 0 0 1 0 0 0 1 1

When $M = \pm 2$, the parameters should be set like the following example:

`TR`n x_0 y_0 z_0 θ_z θ_y θ_x 0 0 0 0 0 0 2

where, x_0, y_0, z_0 are ,respectively, x, y, z-components for the translation, and $\theta_z, \theta_y, \theta_x$ are rotation angles around z, y, x-axes, respectively. For B_4-B_9, any values, such as zero, should be set.

5.7.2 Mathematical definition of transformation

When the coordinates before and after the transformation are defined as (x, y, z) and (x', y', z'), respectively, the mathematical definition is the following except the case of tallies:

for $M = 1$,

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} B_1 & B_4 & B_7 \\ B_2 & B_5 & B_8 \\ B_3 & B_6 & B_9 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} + \begin{pmatrix} O_1 \\ O_2 \\ O_3 \end{pmatrix}, \tag{19}$$

for $M = -1$,

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} B_1 & B_4 & B_7 \\ B_2 & B_5 & B_8 \\ B_3 & B_6 & B_9 \end{pmatrix} \begin{pmatrix} x - O_1 \\ y - O_2 \\ z - O_3 \end{pmatrix}. \tag{20}$$

In the case of $M = 1$, the object using this transform function is rotated and then translated. On the other hand, in the case of $M = -1$, the rotation is performed after the translation. The rotation is performed about the origin of the *xyz* coordinate system. Note that the direction of the translation setting $M = 1$ and -1 is opposite each other.

For the case of tallies, the mathematical definition of $M = 1$ is given as,

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} B_1 & B_2 & B_3 \\ B_4 & B_5 & B_6 \\ B_7 & B_8 & B_9 \end{pmatrix} \begin{pmatrix} x - O_1 \\ y - O_2 \\ z - O_3 \end{pmatrix}, \tag{21}$$

and that of $M = -1$ is given as,

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} B_1 & B_2 & B_3 \\ B_4 & B_5 & B_6 \\ B_7 & B_8 & B_9 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} + \begin{pmatrix} O_1 \\ O_2 \\ O_3 \end{pmatrix}. \tag{22}$$

Because tally results are output in the coordinates system after the transformation, the relation between (x, y, z) and (x', y', z') in the equations in the case of tally is opposite to that in the other case.

When $M = \pm 2$, the input format is simplified. The mathematical definition of $M = 2$ and -2 are the same as $M = 1$ and -1, respectively.

5.7.3 Examples of [transform]

Example 48: `[transform]` section example (1)

```
[ Transform ]
set: c10[90]  $ angle of around Z (degree)
set: c20[30]  $ angle of around Y (degree)
set: c30[0]   $ angle of around X (degree)

tr1   0 0 0
       cos(c10/180*pi)*cos(c20/180*pi)
       sin(c10/180*pi)*cos(c30/180*pi)+cos(c10/180*pi)*sin(c20/180*pi)*sin(c30/180*pi)
       sin(c10/180*pi)*sin(c30/180*pi)-cos(c10/180*pi)*sin(c20/180*pi)*cos(c30/180*pi)
      -sin(c10/180*pi)*cos(c20/180*pi)
       cos(c10/180*pi)*cos(c30/180*pi)-sin(c10/180*pi)*sin(c20/180*pi)*sin(c30/180*pi)
       cos(c10/180*pi)*sin(c30/180*pi)+sin(c10/180*pi)*sin(c20/180*pi)*cos(c30/180*pi)
       sin(c20/180*pi)
      -cos(c20/180*pi)*sin(c30/180*pi)
       cos(c20/180*pi)*cos(c30/180*pi)
       1
```

In this example, `tr1` rotates the coordinate by `c10` degrees around *z* axis, `c20` degrees around *y* axis and finally `c30` degrees around *x* axis. By setting `c10, c20, c30`, the rotation can be defined.

Example 49: `[transform]` section example (2)

```
[ T r a n s f o r m ]
*tr1   0 0 0  90   30   0  0 0 0 0 0 0  2
tr2    0 0 0  pi/2 pi/6 0  0 0 0 0 0 0  2
```

These are examples of $M = 2$, which gives the same definition of the rotation as the above example. The rotation angles of ∗TR1 are given in units of degree, and those of TR2 are given in radian.

5.8 [Temperature] section

Free-Gas Thermal Temperature [MeV] can be defined in this section. This section corresponds to TMP card but the time definition can be set. This value can be set in the `[cell]` section. If the temperature is double defined, temperatures defined in this `[temperature]` sections are used. If no definition, the default value 2.53×10^{-8}MeV is used. Note that in general temperature has an effect on only behavior of thermal neutrons, and has no effect on those of other particles.

```
    [ Temperature ]
        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
        ....             ........
        ....             ........
```

The format `( { 2 - 5 } 8 9 )` can be used. However, any value that is not single numeric must be enclosed value by `(     )` .

The lattice and universe style can be used as `( 6 < 10[1 0 0] < u=3 )`.

To change the order of region number (`reg`) and temperature (`tmp`), set as `tmp reg`. The skip operator `non` can be used.

5.9 [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.

5.10 [Magnetic Field] section

5.10.1 Charged particle

A magnetic field can be set in the PHITS calculation. To use this function for electron and positron, set EGS5 mode.

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 in this section. Setting `imagnf=1` in the `[parameters]` section enables simulation of particle transport in the magnetic field.

```
[ 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 omitted. 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 omitted. 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. These two subroutine are chosen by setting `usrmgt=1, 2` in the `[parameters]` section. For the Wobbler magnet and pulse magnet, `time` means phase of the magnet and starting time, 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 the `[transform]` section.

The format `( { 2 - 5 } 8 9 )` can be used. However, any value that is not single numeric must be enclosed value by `(      )` . The lattice and universe style can be used as `( 6 < 10[1 0 0] < u=3 )`.

By using this format, the different magnetic field can be set 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. The coordinate transformation by `trcl` is needed for different geometrical situation.

When specifying charge number of the projectile particle with `izst` in `[source]` section, the motion of the particle with the number in the magnetic field is described. Using `izst`, *PHITS* can simulate the motion of the particle with charge states. The charge number defined with `izst` doesn't change while the particle moves. It should be noted that particles produced from nuclear reactions are not affected by the value of `izst`; the charge of the produced particle is given as its atomic number.

`[t-track]` uses `deltm` as a step length to describe particle trajectories in a magnetic field that continuouly changes particles' momenta. Please adjust `deltm` to describe the trajectory curve more smoothly.

5.10.2 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 cannot 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.

5.10.3 Implementation of the magnetic field map

Since version 3.10, PHITS can read arbitrary magnetic fields given in the xyz or r-z grid data. The example of `[magnetic field]` for reading the magnetic field map is below:

```
[ Magnetic  Field ]
  reg   typ   gap    mgf   trcl   file
  101   -1    10.0   0.5   0      xyzlist.dat
  102   -2    100.0  3.0   0      rzlist.dat
  103   -3    10.0   10.0  0      xyzmap.dat
  104   -4    1.0    1.0   1      rzmap.dat
```

Four types of magnetic field maps can be read by PHITS, which are specified by typ=$-1, \cdots, -4$ for charged particles, and typ=$-101, \cdots, -104$ for neutrons. Only one magnetic field map can be defined for each type in an input file. The meanings of each type are shown in Table 5.74. (see Sec. 5.10.4 in more detail):

Table 5.74: Available types of magnetic field

typ	explanation
−1 or −101	xyz grid, data list type
−2 or −102	r-z grid, data list type
−3 or −103	xyz grid, data map type
−4 or −104	r-z grid, data map type

The meanings of `gap` and `mgf` are different from those for the case of the conventional magnetic fields, where `gap` is inversely proportional to the step size for calculating particle trajectory, i.e. setting the higher value for gap results in the smoother trajectory but longer computational time, while `mgf` indicates the normalization factor of the field strength for the magnetic field map cases. For example, you have to set `mgf=10` when your magnetic field map is written in the unit of T because the unit of magnetic field in PHITS is kG. Note that electron and positron step sizes in the magnetic field are automatically determined irrelevant to `gap`. Name of the magnetic field map file is specified by file parameter.

5.10.4 Format of magnetic field map file

The magnetic field map file should consist of the header and data parts. In the header part, comment remarks can be inserted using # and $ in similar to the PHITS input file. Examples of the magnetic field map files together with the PHITS input files to read them are included in `\phits\utility\magmap` folder.

Header format

In the header part, discrimination between lowercase and uppercase characters is not performed, and blank is ignored in the same as PHITS input. Only one parameter can be specified in one line. Table 5.75 shows the parameter list to be specified in the header part:

Table 5.75: parameter list in the header part

parameter	explanation
nx	number of x grid (only for typ=-1 & -3, not omissible)
ny	number of y grid (only for typ=-1 & -3, not omissible)
nz	number of z grid (only for typ=-1 & -3, not omissible)
nr	number of r grid (only for typ=-2 & -4, not omissible)
xmin	minimum value of x grid in cm (only for typ=-3, D=0)
xmax	maximum value of x grid in cm (only for typ=-3, not omissible)
ymin	minimum value of y grid in cm (only for typ=-3, D=0)
ymax	maximum value of y grid in cm (only for typ=-3, not omissible)
zmin	minimum value of z grid in cm (only for typ=-3 & -4, D=0)
zmax	maximum value of z grid in cm (only for typ=-3 & -4, not omissible)
rmin	minimum value of r grid in cm (only for typ=-4, D=0)
rmax	maximum value of r grid in cm (only for typ=-4, not omissible)
ibxmap	existence of the B_x map (yes=1, no=0) (only for typ=-3, D=1)
ibymap	existence of the B_y map (yes=1, no=0) (only for typ=-3, D=1)
ibzmap	existence of the B_z map (yes=1, no=0) (only for typ=-3 & -4, D=1)
ibrmap	existence of the B_r map (yes=1, no=0) (only for typ=-4, D=1)
extendx	extend the field to negative x (only for typ=-1 & -3, omissible)
extendy	extend the field to negative y (only for typ=-1 & -3, omissible)
extendz	extend the field to negative z (for all typ, omissible)
data	End of header.

If you want to flip the direction of magnetic field in the extended field, you have to write e.g. `flip=bx, bz`, where `bx` and `bz` indicate the direction of magnetic field to be flipped. For example, if you set:

```
extendx flip = bz
extendy flip = bz
```

the magnetic field for z-direction (B_z) is flipped as shown in the Fig. 5.47. Note that the magnetic fields for x and y directions are not flipped in this case.

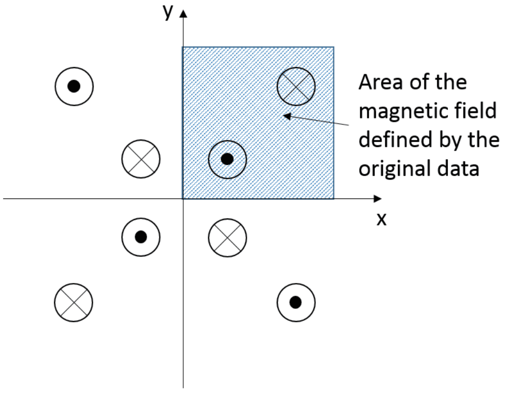


Figure 5.47: Example of the extended magnetic field with the `flip` command

Data format

In the data part, only numerical values with separation of comma, tab, or blank can be written. Comments cannot be inserted except for columns behind the last numerical data of each line. The units of the grid coordinate and the magnetic field are cm and kG, respectively. Note that the magnetic field strength at the point of each grid should be given instead of that at the center of each grid-mesh. The followings are the format of each field type:

- `typ`=-1: xyz grid, data list type

$$\boxed{\begin{array}{cccccc}
x_1 & y_1 & z_1 & B_{x_{1,1,1}} & B_{y_{1,1,1}} & B_{z_{1,1,1}} \\
x_1 & y_1 & z_2 & B_{x_{1,1,2}} & B_{y_{1,1,2}} & B_{z_{1,1,2}} \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
x_1 & y_1 & z_{nz} & B_{x_{1,1,nz}} & B_{y_{1,1,nz}} & B_{z_{1,1,nz}} \\
x_1 & y_2 & z_1 & B_{x_{1,2,1}} & B_{y_{1,2,1}} & B_{z_{1,2,1}} \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
x_1 & y_{ny} & z_{ny} & B_{x_{1,ny,nz}} & B_{y_{1,ny,nz}} & B_{z_{1,ny,nz}} \\
x_2 & y_1 & z_1 & B_{x_{2,1,1}} & B_{y_{2,1,1}} & B_{z_{2,1,1}} \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
x_{nx} & y_{ny} & z_{ny} & B_{x_{nx,ny,nz}} & B_{y_{nx,ny,nz}} & B_{z_{nx,ny,nz}}
\end{array}}$$

- `typ`=-2: r-z grid, data list type

$$\boxed{\begin{array}{cccc}
r_1 & z_1 & B_{r_{1,1}} & B_{z_{1,1}} \\
r_1 & z_2 & B_{r_{1,2}} & B_{z_{1,2}} \\
\vdots & \vdots & \vdots & \\
r_1 & z_{nz} & B_{r_{1,nz}} & B_{z_{1,nz}} \\
r_2 & z_1 & B_{r_{2,1}} & B_{z_{2,1}} \\
\vdots & \vdots & \vdots & \\
r_{nr} & z_{nz} & B_{r_{nr,nz}} & B_{z_{nr,nz}}
\end{array}}$$

- `typ`=-3: xyz grid, data map type

$$\boxed{\begin{array}{ccccc}
B_{x_{1,1,1}} & B_{x_{1,2,1}} & B_{x_{1,3,1}} & \cdots & B_{x_{1,ny,1}} \\
B_{x_{2,1,1}} & B_{x_{2,2,1}} & B_{x_{2,3,1}} & \cdots & B_{x_{2,ny,1}} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
B_{x_{nx,1,1}} & B_{x_{nx,2,1}} & B_{x_{nx,3,1}} & \cdots & B_{x_{nx,ny,1}} \\
B_{x_{1,1,2}} & B_{x_{1,2,2}} & B_{x_{1,3,2}} & \cdots & B_{x_{1,ny,2}} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
B_{x_{nx,1,nz}} & B_{x_{nx,2,nz}} & B_{x_{nx,3,nz}} & \cdots & B_{x_{nx,ny,nz}}
\end{array}}$$

(continue the data for B_y and B_z in the same format)

- `typ`=-4: r-z grid, data map type

$$\begin{array}{ccccc}
B_{r_{1,1}} & B_{r_{1,2}} & B_{r_{1,3}} & \cdots & B_{r_{1,nz}} \\
B_{r_{2,1}} & B_{r_{2,2}} & B_{r_{2,3}} & \cdots & B_{r_{2,nz}} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
B_{r_{nr,1}} & B_{r_{nr,2}} & B_{r_{nr,3}} & \cdots & B_{r_{nr,nz}}
\end{array}$$

(continue the data for B_r in the same format)

The order of the data cannot be changed. However, you can omit to define the field strength for certain directions by specifying `ibxmap, ibymap, ibzmap, ibrmap` for `typ`=-3 & -4. In that case, the field strength for the omitted direction are assumed to be 0. For `typ`=-1 & -2, the grid coordinates should be given in the ascending order. The computational time for the data map types are generally shorter than that for the data list type, but they are nearly equivalent when the interval of each grid coordinate is constant.

5.11 [Electro Magnetic Field] section

Uniform electric and magnetic fields in any region can be set. Defining parameters in this section and setting `ielctf=1` in `[parameters]` section, PHITS can simulate a motion of a charged particle in the fields. To use this function for electron and positron, set EGS5 mode.

Both the electric and magnetic fields can be set. It is noted that a quadrupole magnet cannot be defined, unlike the case of `[Magnetic Field]` section.

Cell number (`reg`), strength of the electric and magnetic fields (`elf` and `mgf`, respectively), direction of the two fields (`trcle` and `trclm`) should be defined. Units of `elf` and `mgf` are kV/cm and kGauss, respectively. The coordinate transformation number `trcl`, which are defined in `[transform]` section, should be set to `trcle` and `trclm`. When `trcle` is 0, the direction of the electric field is positive direction of *x*-axis. When `trclm` is 0, the direction of the magnetic field is positive direction of *y*-axis. `trcle` and `trclm` are not omissible. In the case that there is no need to When setting `elf` or `mgf` to 0, or not using the transformation, `trcle` and `trclm` should be set to 0. Set the parameters as follows.

```
[ Electro Magnetic Field ]
  reg    elf    mgf    trcle    trclm
    1    100      1        1        2
    2    100      1        1        2
```

When specifying charge number of the projectile particle with `izst` in `[source]` section, the motion of the particle with the number in the electro-magnetic field is described. Using `izst`, PHITS can simulate the motion of the particle with charge states. The charge number defined with `izst` doesn't change while the particle moves. It should be noted that particles produced from nuclear reactions are not affected by the value of `izst`; the charge of the produced particle is given as its atomic number.

`[t-track]` uses `deltm` as a step length to describe particle trajectories in an electro-magnetic field that continuouly changes particles' momenta. Please adjust `deltm` to describe the trajectory curve more smoothly.

5.12 [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, [48] considering the relativistic collision dynamics.

This function shouldn't be used together with `[t-sed]` tally.

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.

[48] J. J. Butts and R. Katz, "Theory of RBE for Heavy Ion Bombardment of Dry Enzymes and Viruses", Radiation Research 30, 855-871 (1967).

5.13 [Track Structure] section

Performing the track-structure simulation [49] for electrons or positrons of low energy below 1 keV, those deceleration processes caused by much collision events for ionization, excitation, and molecular vibration and rotation can be calculated precisely. Currently, the cross sections only for liquid water are prepared in PHITS, and those for other materials are simply obtained from electron density scaling based on the data. The track-structure simulation takes so long time that we cannot recommend to activate this mode in a conventional-scale particle transport simulation (cm orders) due to much production of low energy electrons around 10 eV. Please take cares that the track-structure mode can be adapted only to `[t-track]` and `[t-deposit]` tallies.

In this section, you have to specify the cells where you would like to perform track-structure simulation. `mID` represents the index of the cross sections database used in the track-structure simulation, and currently you can select only 0 (no track structure simulation) or 1 (track structure simulation using the database of liquid water). If you will activate `mID = 1`, mean free paths of the particles in target materials are derived from electron density scaling based on the database for liquid water. We are planning to prepare the cross section databases for other materials in future.

```
[track structure]
reg  mID
  1    1
  2    0
```

You can use the format `(   2 - 5   89 )`. 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[100] < u=3 )`. If you want to replace the order of region number (`reg`) and index of the cross section database (`mID`), set as `mID reg`. You can use the skip operator `non`.

The followings are the important parameters for this mode. The parameters of `etsmax` and `etsmin` in `[parameters]` section are maximum and minimum energies of particles simulated by track-structure mode. In case of the use of the track-structure mode, the parameters of `emin(12)` and `emin(13)` should be set concurrently to 1.0e-3, and EGS5 should be activated (`negs`=1).

```
[ Parameters ]
emin(12) = 1.E-03
emin(13) = 1.E-03
negs     = 1
etsmax   = 1.E-2
etsmin   = 1.E-6
```

[49] T. Kai et al., “Thermal equilibrium and prehydration processes of electrons injected into liquid water calculated by dynamic Monte Carlo method,” Radiat. Phys. Chem., 115, 1-5 (2015).

5.14 [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.

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

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.

5.15 [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 cannot 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.

5.16 [Data Max] section

`dmax(i)`, which is the maximum energy of library use for i-th particle, can be defined for each nucleus in materials in this section. Maximum 6 `[Data Max]` sections are allowed to be used in a input file.

```
[Data Max]
  part = neutron proton
  mat   nucleus  dmax
   all   Fe      20
    5    all     50
    3    56Fe    150
```

Particle is defined in the first line as `part=`. Only neutron and proton can be define in PHITS ver. 2.86.

Three columns, `mat, nucleus`, and `dmax`, can be used. If you want to change the order of (mat) (nucleus), set as `nucleus mat`. You can use the skip operator non. In the mat column, material numbers can be specified and `all` means all materials. In the nucleus column, you can specify nucleus as 56Fe and 26056 type. You can use Fe or 26000, which specifies all isotopes of Fe. You can also use all. By the number (MeV) in the `dmax` column, you define the maximum energy of library use for the nucleus.

If the same nucleus is defined in the `[data max]` sections, the latest definition has priority in an input file.

The values of `dmax(1)` and `dmax(2)` defined by `[parameters]` section should be the maximum value in `[data max]` section.

If `kmout=1` is specified in `[parameters]` section, the values of dmax for each nucleus in the materials are shown in the output file, `file(6) (D=phits.out)`.

5.17 [Frag Data] section

The function to read user defined cross sections can be defined in the `[frag data]` section. By using this function, you can simulate nuclear reaction events in the calculation on the basis of information given by the cross section data. It is noted that nuclear data libraries are preferentially used, even in the case of a nuclear reaction you specify in this section. You can set only proton, neutron, and nuclei as incident particles. Photons and elections cannot be used. Elastic scattering events for nucleons are separately considered by the default model implemented in PHITS.

Because weight values of the secondary particles produced by this option change from 1.0, you should decrease the value of the cutoff weight parameter `wc2(i)` in the `[parameters]` section.

Set the parameters as follows.

```
[ Frag Data ]
  opt     proj     targ     file
    0     12C      16O      DDX_12C-16O.dat
    1     proton   63Cu     DDX_p-63Cu.dat
  ....     ...      ...      ...      ...
  ....     ...      ...      ...      ...
```

The user defined cross sections are not used when `opt=0`. In the case of `opt=1`, PHITS reproduces the cross section of the reaction between an incident particle and target, which are specified by `proj` and `targ`, respectively, using the data. Options when `opt=2,3` are under construction. When `opt=4`, PHITS simply extrapolates the given data for incident energies, emission angles, and emission energies. Note that you can use this option only in the case of neo>0 and nag≠0, which will be explained below.

Format of the data of the user defined cross sections is as follows.

```
projectile
target
nei
ein(1)    ein(2)    ein(3)    ...... ein(nei+1)
totxs(1)  totxs(2)  totxs(3) ...... totxs(nei+1)
neo
eout(1)  eout(2)  eout(3) ...... eout(neo+1)
nag
angle(1)  angle(2)  angle(3) ...... angle(nag+1)
nfrg
frag(1)   frag(2)   frag(3) ...... frag(nfrg)

proxs(1,1)   proxs(1,2)    proxs(1,3) ...... proxs(1,nfrg)
ddx(1,1,1,1)   ddx(1,1,1,2)   ddx(1,1,1,3) ...... ddx(1,1,1,nag)
ddx(1,1,2,1)   ddx(1,1,2,2)   ddx(1,1,2,3) ...... ddx(1,1,2,nag)
       .........
ddx(1,1,neo,1)   ddx(1,1,neo,2)   ddx(1,1,neo,3) ...... ddx(1,1,neo,nag)

ddx(1,2,1,1)   ddx(1,2,1,2)   ddx(1,2,1,3) ...... ddx(1,2,1,nag)
ddx(1,2,2,1)   ddx(1,2,2,2)   ddx(1,2,2,3) ...... ddx(1,2,2,nag)
       .........
ddx(1,2,neo,1)   ddx(1,2,neo,2)   ddx(1,2,neo,3) ...... ddx(1,2,neo,nag)

       .........
       .........
```

（continued）

```
ddx(1,nfrg,1,1)   ddx(1,nfrg,1,2)   ddx(1,nfrg,1,3) ...... ddx(1,nfrg,1,nag)
ddx(1,nfrg,2,1)   ddx(1,nfrg,2,2)   ddx(1,nfrg,2,3) ...... ddx(1,nfrg,2,nag)
      .........
ddx(1,nfrg,neo,1)   ddx(1,nfrg,neo,2)   ddx(1,nfrg,neo,3) ...... ddx(1,nfrg,neo,nag)

proxs(2,1)   proxs(2,2)    proxs(2,3) ...... proxs(2,nfrg)
ddx(2,1,1,1)   ddx(2,1,1,2)   ddx(2,1,1,3) ...... ddx(2,1,1,nag)
ddx(2,1,2,1)   ddx(2,1,2,2)   ddx(2,1,2,3) ...... ddx(2,1,2,nag)
      .........
ddx(2,1,neo,1)   ddx(2,1,neo,2)   ddx(2,1,neo,3) ...... ddx(2,1,neo,nag)


      .........
      .........
      .........
      .........

proxs(nei+1,1)   proxs(nei+1,2)    proxs(nei+1,3) ...... proxs(nei+1,nfrg)
ddx(nei+1,1,1,1)   ddx(nei+1,1,1,2)   ddx(nei+1,1,1,3) ...... ddx(nei+1,1,1,nag)
ddx(nei+1,1,2,1)   ddx(nei+1,1,2,2)   ddx(nei+1,1,2,3) ...... ddx(nei+1,1,2,nag)
      .........
ddx(nei+1,1,neo,1)   ddx(nei+1,1,neo,2)   ddx(nei+1,1,neo,3) ...... ddx(nei+1,1,neo,nag)

      .........
      .........

ddx(nei+1,nfrg,1,1)   ddx(nei+1,nfrg,1,2)   ddx(nei+1,nfrg,1,3) ...... ddx(nei+1,nfrg,1,nag)
ddx(nei+1,nfrg,2,1)   ddx(nei+1,nfrg,2,2)   ddx(nei+1,nfrg,2,3) ...... ddx(nei+1,nfrg,2,nag)
      .........
ddx(nei+1,nfrg,neo,1)  ddx(nei+1,nfrg,neo,2)  ddx(nei+1,nfrg,neo,3) ...... ddx(nei+1,nfrg,neo,nag)
```

In the beginning, you specify incident particle and target in this file. nei is the number of points of the energy mesh. In the next line, you set nei+1 points of incident energies (ein in the unit of MeV/u). neo is the number of energy mesh points of the outgoing particles. In the next line, you set neo+1 points of the energy (eout in the unit of MeV/u), and then in the next next line you set neo+1 data of total reaction cross sections (totxs in the unit of mb). Note that when totxs=0, total reaction cross sections obtained by models, which are specified by `icxsni` or `icrhi`, are used. nag is the number of angular mesh points for the outgoing particles. You set nag+1 data of the angles in the next line. If nag>0 the data should be given in the unit of radian, and if nag<0 those are in the unit is degree. When nag=0, isotropic is assumed. nfrg is the number of the outgoing particles. In the next line, you write nfrg names of the particles. proxs (in the unit of mb) is the production cross sections of the particles at an incident energy, and then you set neo×nag data of the double differential cross sections (ddx in the unit of mb/MeV/sr). You have to specify nei+1 groups of the proxs and ddxs per an outgoing particle.

If you write `model` in the line where you should write neo, nuclear reaction models are used for simulating nuclear reaction events. In this case, you do not have to write the data below neo.

Example 50: example of data file of user defined cross sections

```
proton
63Cu
1
10.0    100.0
1000.0  500.0
3
1.0  10.0  50.0  100.0
-6
0.0  30.0  60.0  90.0  120.0  150.0  180.0
```

```
10:   1
11:   neutron
12:   300.0
13:   10.0  10.0  10.0  10.0  10.0  10.0
14:   15.0  13.0  12.0  11.0  10.0  10.0
15:   10.0  11.0  10.0  11.0  10.0  10.0
16:   0.0
17:    5.0   5.0   5.0   5.0   5.0   5.0
18:   10.0   8.0   7.0   6.0   5.0   5.0
19:    5.0   6.0   5.0   6.0   5.0   5.0
```

An example of the data file of the user defined cross sections is shown in Example 50. In first and second lines, proton and ^{63}Cu are defined as incident particle and target, respectively. In third line, nei=1 is set. Tow data of incident energies and total reaction cross sections are given in fourth and fifthe lines, respectively. In sixth line neo=3 is set, and then 4 values of 1.0, 10.0, 50.0, and 100.0MeV are given as energies of the outgoing particles. The number of angular mesh points nag=-6 is set in eighth line. In this case, the angles are given in the unit of degree. In tenth and eleventh lines, the number of particles and its kind (neutron) are specified, respectively. In twelfth line, a production cross section of neutron at the 10.0MeV proton is given. The double differential cross sections of the neutron are set in thirteenth, fourteenth, and fifteenth lines. Each lines correspond to the energy bins defined in seventh line, and each columns correspond to the angular bins defined in ninth line. The production cross section is used as a normalization factor. When proxs=0 (mb), the integrated values of the ddxs are used as proxs.

When neo is negative, you can set the intensity of the cross section discretely. In this case, the number of eout should be neo.

If neo is 0, the energy spectra of the outgoing particle can be given by Gaussian distribution. In the places of ddxs, you set mean value and FWHM of the Gaussian in the unit of MeV.

5.18 [Importance] section

The importance for GG cell can be defined in this section. If the importance is not defined, it is set as `1.0`. Maximum 6 `[importance]` sections are allowed to be defined in an input file.

```
    [ Importance ]
       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

```
[ Importance ]
   reg            imp
 ( 5 6 7 < 11 )   2.0
 ( 5 6 7 < 12 )   4.0
 ( 5 6 7 < 13 )   8.0
 ( 11 12 13 )     1.0
```

or

```
[ Importance ]
   reg            imp
 ( 5 6 7 )        1.0
   11             2.0
   12             4.0
   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.

5.19 [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 ]
            mesh =  reg
            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
                            ........    ........
```

Mesh type should be defined in the 1st line, i.e. `mesh=reg` or `xyz`. When `mesh` is not defined, the default setting `mesh=reg` is set. For `mesh=xyz`, `x-type, y-type`, and `z-type` must be defined in subsequent lines (see Sec. 6. Geometrical mesh in detail). Particle type should be defined 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 (`ref` or `xyz`) 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.

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. For `mesh = xyz`, the position of each data should be defined as `(ix iy iz)`.

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.

5.20 [WW Bias] section

In the `[ww bias]` section, minimum values of the weight window defined in `[weight window]` can be biased for certain regions. This option is useful when the `[weight window]` parameters for a certain region are biased after automatically generating the `[weight window]` section by `[t-wwg]`. Figure 5.48 shows a flowchart how to perform the transport calculation using `[weight window]` and `[ww bias]`. First, generate the `[weight window]` section by `[t-wwg]`. If the obtained parameters of `[weight window]` are enough to effectively use the variance reduction technique, the `[ww bias]` section is not needed. However, if they are not enough, the variance reduction technique can be used more effectively by biasing the `[weight window]` parameters for a certain region with `[ww bias]`. There are two methods to set the `[ww bias]` section; an automatic method using `[t-wwbg]`, which is a Weight Window Bias Generator (WWBG), and a manual method by a user, which will be explained below. For details of `[t-wwbg]`, see Sec. 7.16. When performing a transport calculation with `[ww bias]` and `[weight window]`, set `icntl=0` and `iwwbias=1` in `[parameters]`.

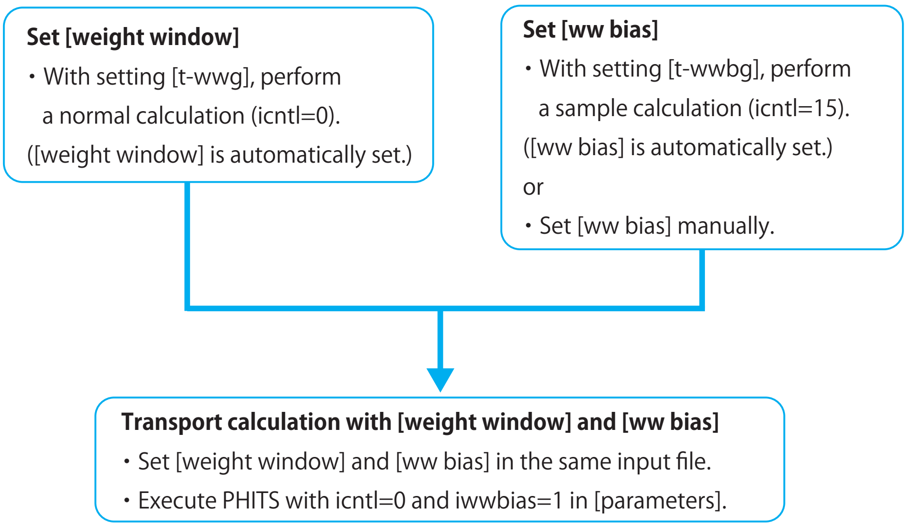

Set [weight window]
• With setting [t-wwg], perform
a normal calculation (icntl=0).
([weight window] is automatically set.)
Set [ww bias]
• With setting [t-wwbg], perform
a sample calculation (icntl=15).
([ww bias] is automatically set.)
or
• Set [ww bias] manually.
Transport calculation with [weight window] and [ww bias]
• Set [weight window] and [ww bias] in the same input file.
• Execute PHITS with icntl=0 and iwwbias=1 in [parameters].

Figure 5.48: The flowchart of the connection calculation between `[weight window]` and `[ww bias]`.

The format of `[ww bias]` is as follows: (Note that set the same particle, energy-mesh, and cell numbers as `[weight window]`.)

```
        [ WW Bias ]
           part =  neutron
            eng =  2
                   1e-3   1.0
            reg    wwb1        wwb2
              1    0.25        0.25
              2    0.50        0.50
              3    1.00        1.00
              4    2.00        2.00
           ....    ........    ........
```

In the first line, `part=` defines which particle is to be considered. When it is omitted, `part=all` is set. The expression of `part=` is the same as that in the tally format. Note that only the expression as `ityp` can be set. Each nuclides cannot be specified. Next, the energy mesh should be defined. The line starting with `eng=` specifies the number of mesh. In the next line, energies ($e_1, e_2, e_3, ...$) are defined. Furthermore, names of columns are gives as `reg, wwb1, wwb2, ....` In the `reg` column, the cell numbers are written. The bias values are given in the columns of `wwb`*i*. The skip operator `non` can be used. Each `wwb`*i* column corresponds to energies of $e_{i-1} < E < e_i$.

Here, $e_0 = 0$. The format `( { 2 - 5 } 8 9 )` can be used, as can the lattice and universe style `( 6 < 10[1 0 0] < u=3 )`. However, any value that is not single numeric must be enclosed value by `(    )` .

By setting `iwwbias=1` in `[parameters]`, the `[weight window]` parameters multiplied by inverse of the defined biases in `[ww bias]` are used. In this case, the products of the multiplication are output in the input echo of `[weight window]`, and `[ww bias]` with `off` is output. If an input file without `[ww bias]` is used, all values of `[ww bias]` in the input echo are set to 1.

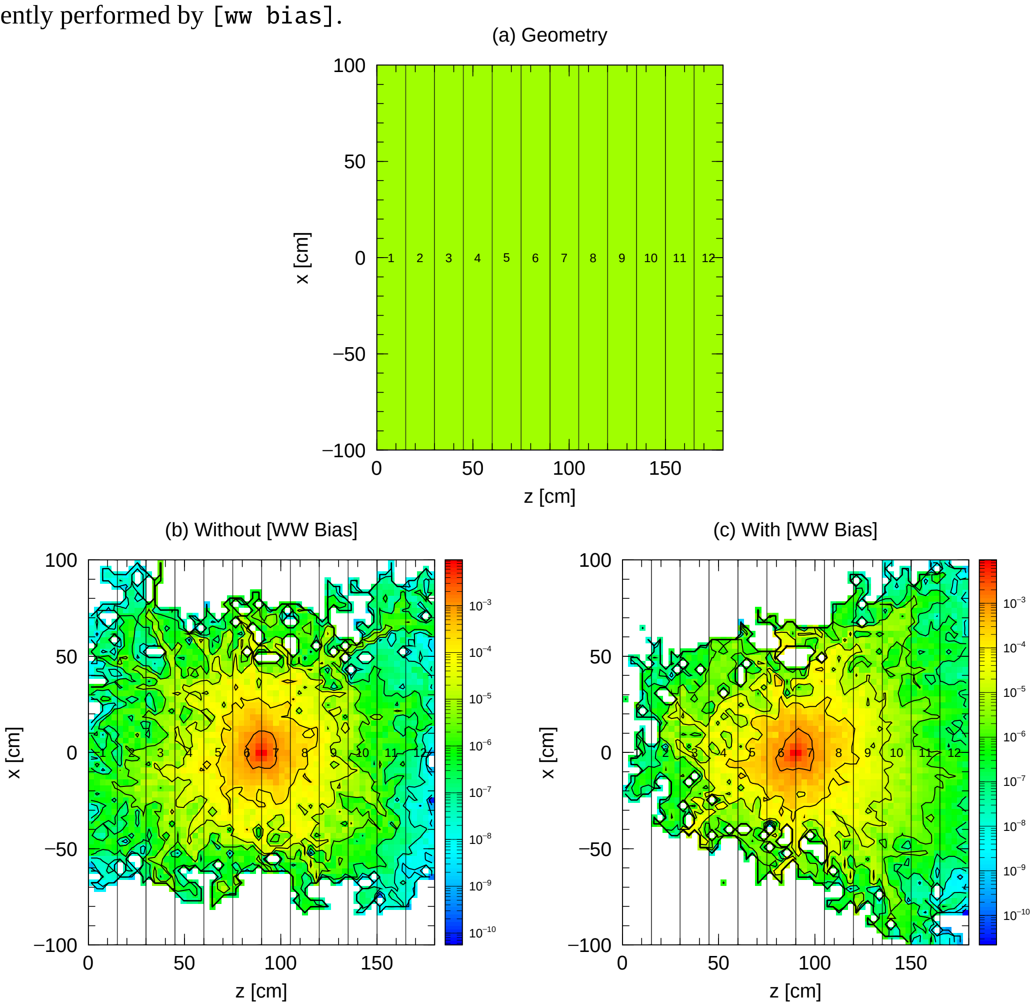
An example of `[ww bias]` is as follows.

Example 51: Example of `[ww bias]`

```
[ WW Bias ]
   part = neutron
    eng =  1
         1.00000E+05

   reg       wwb1
   1         1/7
   2         1/6
   3         1/5
   4         1/4
   5         1/3
   6         1/2
   7         1
   8         2
   9         3
   10        4
   11        5
   12        6
```

Here, neutron is considered as `part`. One energy region of 100 GeV below is specified. Regions between 1 and 12 are gradually biased. The region of the large number is biased stronger than that of the small number.

Figure 5.49(a) shows the *xz* cross-section view of a geometry; a concrete cylinder with a central axis on the *z* axis with a radius of 100cm. Two results of the neutron fluence obtained by the transport calculation without and with `[ww bias]` were shown in Figs. (b) and (c), respectively. Source particles of 14MeV-neutrons were generated at $x = 0, y = 0, z = 90$cm as an isotropic source. After generating a `[weight window]` section by `[t-wwg]`, the result shown in Fig. (b) was obtained by performing the calculation with only the `[weight window]` section. The result of the calculation with both the `[weight window]` section and the `[ww bias]` section of the example 51 was shown in Fig. (c). The neutron fluence in Fig. (b) was distributed in both regions of the small and large cell numbers. On the other hand, the fluence in Fig. (c) was distributed in only the large cell numbers, which were biased by `[ww bias]` of the example 51. As seen in this example, to focus on a certain region, the calculation can be efficiently performed by `[ww bias]`.


(a) Geometry

(b) Without [WW Bias]

(c) With [WW Bias]

Figure 5.49: (a) *xz* cross-section view of the geometry. (b) Result without `[ww bias]`. (c) Result with `[ww bias]` of Example 51.

5.21 [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 collision point is randomly determined according to cross sections. 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 an 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 to control the particle transport and multiple scattering with the weight cut off in the forced collisions 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. For example, it corresponds to tallying tracks of secondary particles and information of particles at a distance from the forced collisions region. When you want to transport secondary particles produced by forced collisions, you should decrease the value of the cutoff weight parameter `wc1(i)` in the `[parameters]` section.

5.22 [Volume] section

Volume for GG cell (cm^3) can be defined in this section. If the volume is double defined, the value defined in this section is used. The volume is utilized in the tally sections. If you do not set volume, it has 1.0 cm^3 volume.

```
   [ Volume ]
       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 cannot 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.

5.23 [Multiplier] section

In the [multiplier] section, multiplier sets consisting of factors that depend on particle energies can be set as multipliers of [t-track], [t-cross], [t-point] tally results. To use this function, the multiplier subsections in the tally section must be defined. For example, this function can be used for dose estimation using any dose conversion factor.

Only one multiplier set can be defined per [multiplier] section; a maximum of 100 [multiplier] sections that can be defined in an input file. The format of this section is given as follows.

```
[ Multiplier ]
  number = -201
  interpolation = log
  part = all
  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 each [multiplier] section can be set by negative integer between -200 and -299. Particle type can be specified by part parameter. part = all indicates that the data are applicable to all particle. For the data interpolation method, you can select from lin, log, glow, and ghigh, which indicate the linear-linear interpolation, the log-log interpolation, and group data by specifying lower and higher boundary of each energy group, respectively. The number of the energy point or group is given by ne, and the data of each energy and factor should be defined in the subsequent lines. Note that the data should be listed in ascending order of energy.

From ver. 3.03, the conversion coefficients for several types of radiation doses and soft error rate (SER) on semiconductor devices are pre-defined. Table 5.76 shows the multiplier ID of pre-defined data. Those data are included in phits/data/multiplier directory with the file name of “m+|ID|+.inp” (e.g., m200.inp for $k = -200$). The unit of conversion coefficients for radiation dose is pSv·cm^2, and that for soft error rate is (FIT/Mbit)/(/cm^2/s). You can add your own [multiplier] section in this folder. If you specify [multiplier] section with the same ID number in your input file, the pre-defined data are overwritten by the specified data.

The conversion coefficients for SER are derived from the neutron-induced SEU cross section for a virtual semiconductor device calculated by PHITS and device simulator [50]. The background SER[51] is about 400 (FIT/Mbit). Here, 1 (FIT) = 1e-9 (error/hour). Generally, SEU cross sections depend on the device. Therefore, the conversion coefficients are not used to estimate the exact SER. Please utilize them to estimate SER roughly or to compare SER in the radiation field and background SER.

[50] The critical charge is assumed to be 0.6 fC. The neutron energies from 1 MeV to 1 GeV are considered. For details, please refer to “S. Abe and Y. Watanabe, IEEE Trans. Nucl. Sci. 61, 3519-3526 (2014).”

[51] The background means on the ground at Tokyo. The cosmic-ray neutron flux is calculated by PARMA model. The PARMA model can be downloaded from http://phits.jaea.go.jp/expacs

Table 5.76: Pre-defined conversion coefficients.

Multiplier ID	Data information
-200	$H^*(10)$[52]
-201	Effective dose based on ICRP60 (AP irradiation)[53]
-202	Effective dose based on ICRP103 (AP irradiation)[54]
-203	Effective dose based on ICRP103 (ISO irradiation)[54]
-204	New operational quantity H^* (Maximum effective dose among the all irradiation conditions[54]
-210	Sex-averaged effective dose equivalent (ISO irradiation)[55]
-211	Effective dose equivalent for male (ISO irradiation)[55]
-212	Effective dose equivalent for female (ISO irradiation)[55]
-213	Dose equivalent for male red-bone marrow (ISO irradiation)[55]
-214	Dose equivalent for female red-bone marrow (ISO irradiation)[55]
-215	Dose equivalent for male skin (ISO irradiation)[55]
-216	Dose equivalent for female skin (ISO irradiation)[55]
-299	Soft error rate on semiconductor device (ISO irradiation)[56]

5.23.1 Multiplier subsection

The multiplier option of the tally section is used to define a multiplier set following the basic format ($C \quad k$), where C is a normalization factor and k is the ID number of the set (note that k should be negative). The 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 starting with `multiplier`= specifies the number of materials for which multiplication is considered; `all` can be used instead of a number, in which case all should be used for the following `mat` column. In the second line, `part`= defines which particle is to be considered. A maximum of six particles can be entered here; `all`, which is the default, can be also used. Only those particles listed will be multiplied. In the third line, `emax`= defines the maximum energy of multiplication. If `emax` is omitted, it is automatically defined as the maximum energy given in the `[multiplier]` section. The numbers of the respective mat columns give the material numbers considered for multiplication. The columns `mset1` and `mset2` define the multiplier sets up to a maximum of six sets. The result of each set is printed out. Although several multiplier subsections can be defined in one tally section, the number of multiplier sets should be constant across subsections.

PHITS has some built-in coefficients available for the `[multiplier]` section. If you set $k = -1$, a value of 1/weight is used as the multiplication factor to obtain tally results of a Monte Carlo particle, i.e., particles always having their weights as 1. For $k = -2$, a value of 1/velocity is used. For $k = -120$, material density is used. Therefore, you can obtain mass in the region setting `icntl=5`. You can also set the ID number included in

[52] Taken from EXPACS (http://phits.jaea.go.jp/expacs/)

[53] Y. Sakamoto, O. Sato, S. Tsuda, N. Yoshizawa, S. Iwai, S. Tanaka, and Y. Yamaguchi, “Dose conversion coefficients for high-energy photons, electrons, neutrons and protons”, JAERI-1345, (2003) etc.

[54] ICRP Publication 116, Ann. ICRP 40(2-5), 2010.

[55] ICRP Publication 123, Ann. ICRP 42(4), 2013.

[56] S. Abe and Y. Watanabe, IEEE Trans. Nucl. Sci. 61, 3519-3526 (2014).

/phits/data/multiplier/ directory without specifying the [multiplier] section in the input file. Currently, the databases of the conversion coefficients for several types of radiation doses are included in the directory (see Table 5.76 in detail). The unit of the dose conversion coefficients is pSv.cm^2, and thus, the doses in the unit of pSv/sec can be directly calculated when the calculated fluence is normalized to the unit of /cm^2/sec. Note that the energy unit of the dose conversion coefficients for heavy ions is MeV/u, and you have to set iMeVperu = 1 in the [parameters] section when you use the heavy ion data in your simulation.

In addition to them, the effective doses based on ICRP60 for the AP irradiation from proton, neutron, electron, and photon can be separately calculated by specifying $k = -101, -102, -112$, and -114, respectively, though this method is not recommended anymore because the corresponding doses can be calculated by $k = -201$. Note that the unit of these dose conversion coefficients is (μSv/h)/(n/sec/cm^2), i.e. the doses in the unit of μSv/h can be directly calculated when the calculated fluence is normalized to the unit of /cm^2/sec. 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.

```
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.

5.24 [Mat Name Color] section

Material names, size and colors for graphical output by `[t-gshow]` and `[t-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
           ....            ........
           ....            ........
```

To replace the order of material number (`mat`), (`name`), (`size`), and (`color`), set as `mat color size name`. The skip operator `non` can be used. At least one parameter in `name` and `color` must be defined. If no definition, the default values are used.

The format { `4 - 7` } can be used, but the format `( { 4 - 7 } 9 10 )` cannot be used. To use blanks in name definition, the name must be closed by { } as the example. To use `( )`, the format `\( \)` should be used. In the name, { } cannot be used. Note that a superscript or subscript in the LaTeX format can be used by writing \{ \}. For example, in the case of writing ˆ\{ `208` \}`Pb`, ^{208}Pb is output. The maximum number of characters to define a name is 80.

When defining two (or more) cells of different densities from each other with the same material number in `[cell]` section, the cells except the first one are given other material numbers in `matadd=1` (the default setting). The given numbers are written in the first part of `file(6) (D=phits.out)` as a warning message. Therefore, set the numbers have to be set in the `mat` column of this section. Even if setting `matadd=0` in `[parameters]` section, the function unifying the material number is invalid in this section.

The color definition is based on the format in ANGEL. 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 5.77: Gray scale

Symbol	HSB	Output	Name
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 5.78: Color definition by symbols

Symbol	HSB	Output	Name
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 5.79: Color definition by names and HSB numerics

Name	Output	HSB		
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

5.25 [Reg Name] section

Region names and their font sizes for graphic output by `[t-gshow]`, `[t-rshow]`, and `[t-3dshow]` are defined in this section. When you set the gshow or rshow option in the other tallies, this definition is applied. By default, a region name is set to its 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`), region name (`name`), and font size (`size`), set as `reg size name`. You can use the skip operator `non`. At least one of `name` and `size` must be defined. If nothing is defined, it is assumed to be default. You can use the format { `4 - 7` }, but the `(` { `4 - 7` } `9 10 )` format cannot be used. If you need to use blanks in the name definition, the name must be closed by { } as the example. If you want to use `(` and `)`, you should write `\(` and `\)`, respectively. Brackets { and } cannot be used in the name definition. Note that a superscript or subscript in the LaTeX format can be used by writing \{ \}. For example, in the case of writing `^`\{ `208` \}`Pb`, ^{208}Pb is output. The maximum number of characters of a name that you can define is 80. You can specify a font size as a relative value to the default size.

5.26 [Counter] section

The counter function can be defined in this section. Three counters can be used in tally sections. The counter basically counts when

(1) a particle comes into specified region (in)

(2) a particle goes out specified region (out)

(3) a particle takes scattering (or nuclear reaction) in specified region (coll)

(4) a particle reflects back on a certain boundary of the region (ref)

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 collisions inherit the counter value of their 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.

```
[ Counter ]
   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 latter case, `Pb`, denotes all isotopes of `Pb`.

From ver. 2.90, by detailed classification for `coll`, the opportunities shown in Table 5.80 are available as keywords so that the counter counts. `nucl, atom`, and `dcay`, which are particular interactions, belong to `coll`. Furthermore, `nucl` and `atom` are classified into three and ten kinds, respectively. When you want to analyze the PHITS simulation in detail, set these keywords. Note that if you set `coll` and `nucl` at the same time, the counting is duplicated when a nuclear reaction event occurs.

When (`fiss`) is specified, the counter is called when fission channels of the nuclear data library and the statistical decay model (GEM) are chosen.

Table 5.80: Categories of reaction events belonging `coll`.

Interaction	Event	Explanation
`nucl`		A particle takes nuclear interactions in specified region.
	`elst`	A particle takes elastic scattering in specified region.
	`iels`	A particle takes inelastic scattering in specified region.
	`fiss`	A particle takes fission in specified region.
`atom`		A particle takes atomic interactions and generate secondary particle in specified region.
	`delt`	A particle generates delta-rays in specified region.
	`knoe`	An electron or positron generates knock-on electrons in specified region.
	`fluo`	A particle generates atomic fluorescence x-rays in specified region.
	`auge`	A particle generates Auger electrons in specified region.
	`brem`	A particle takes bremsstrahlung in specified region.
	`phel`	A particle takes photoelectric effect in specified region.
	`cmpt`	A particle takes Compton scattering in specified region.
	`pprd`	A particle generates electron-positron pair in specified region.
	`anih`	A particle is annihilated in specified region.
	`rayl`	A particle takes Rayleigh scattering in specified region.
`dcay`		A particle decays in specified region.

5.27 [Timer] 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).

```
   [ Timer ]
       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.

6 Common parameters for tallies

PHITS has the following tally functions.

Table 6.1: Tally sections

name	explanation
[t-track]	Particle fluence in a certain region.
[t-cross]	Particle fluence crossing at a certain surface.
[t-point]	Particle fluence at a certain point.
[t-deposit]	Deposit energy in a certain region.
[t-deposit2]	Deposit energies in certain two regions.
[t-heat]	Heat generation in a certain region. (Not recommended[57])
[t-yield]	Residual nuclei yield in a certain region.
[t-product]	Produced particle in a certain region.
[t-dpa]	Displacement Per Atom (DPA) in a certain region.
[t-let]	LET distribution in a certain region.
[t-sed]	Microdosimetric quantity distribution in a certain region.
[t-time]	Time information of particle in a certain region.
[t-interact] (formerly named [t-star])	Number of interactions occurred in specified regions.
[t-dchain]	Residual nuclide yields (in combination with DCHAIN).
[t-wwg]	Output parameters for [weight window].
[t-wwbg]	Output parameters for [ww bias].
[t-volume]	Automatic calculation of region volume.
[t-userdefined]	Any quantities that user defined.
[t-gshow]	2D geometry visualization.
[t-rshow]	2D geometry visualization with physical quantities.
[t-3dshow]	3D geometry visualization.

Common parameters used in these tallies are described below.

6.1 Geometrical mesh

In the tallies shown by Table 6.1, GG region mesh (reg), r-z scoring mesh (r-z), and xyz scoring mesh (xyz) can be used for geometrical mesh of tallying area. Additionally tetrahedral mesh (tet) can be used for [t-track], [t-deposit], [t-yield], [t-product], and [t-dpa].

You can choose one mesh from:

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

6.1.1 Region mesh

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

[57] Before ver. 3.04, the [t-heat] tally was used to calculate deposit energy using the kerma approximation, because the [t-deposit] tally did not have the option.

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

Each cell number should be separated by blank. Some regions can be combined by using `(    )`. 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 cannot specify like `( n1 - n2 )`. Styles `( { } )` and `( all )` can be used, but `{ ( ) }` cannot 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. Furthermore, you set region as `reg = all`, all regions become tallying region. However, cells which do not belong to bottom level, are not included.

6.1.2 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. You can also 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.

Example 52: `mesh=reg` example (1)

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

This region mesh definition is echoed as

Example 53: `mesh=reg` example (2)

```
mesh =  reg            # mesh type is region-wise
 reg = ( all ) ( { 201 - 205 } ) ( 161 < 160[ 1:2 3:6 1:1 ] ) ( (
       { 201 - 204 } ) < ( { 161 - 163 } ) ) ( ( 90 100 ) 120 < 61
       ( 62 63 ) )
volume                 # combined, lattice or level structure
non   reg      vol     # reg definition
 1  10001   8.1000E+01 # ( all )
 2  10002   5.0000E+00 # ( { 201 - 205 } )
 3  10003   1.0000E+00 # ( 161 < 160[ 1 3 1 ] )
 4  10004   1.0000E+00 # ( 161 < 160[ 2 3 1 ] )
 5  10005   1.0000E+00 # ( 161 < 160[ 1 4 1 ] )
 6  10006   1.0000E+00 # ( 161 < 160[ 2 4 1 ] )
 7  10007   1.0000E+00 # ( 161 < 160[ 1 5 1 ] )
 8  10008   1.0000E+00 # ( 161 < 160[ 2 5 1 ] )
 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.

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

6.1.3 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 omitted. Then, define r and z mesh as

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

Mesh definition is described later.

You can add θ degree of freedom in r-z mesh of `[t-track]` tally. You can use `a-type` mesh definition to specify θ of r-z mesh in the same manner as in the other angle mesh cases. However, the maximum angle of θ mesh is restricted 360 degree (2π) in this case. You can obtain the θ dependence of results of `[t-track]` tally by setting `axis=rad` or `axis=deg`. The `rad` represents the θ in radian, while `deg` in degree.

6.1.4 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.

6.1.5 tet mesh

This mesh is applicable only when tetrahedron geometry5.6.5 is used in the geometry and only active in [t-track], [t-deposit], [t-yield], [t-product], [t-dpa] tallies. When you use the tet scoring mesh, a cell no. of the rectangular box where the tetrahedron geometry is implemented needs to be specified following to the mesh definition as

```
        mesh   = xyz
         reg   = 100
```

The physical quantities specfied in the tally will be scored in all the tetrahedrons separatelly. Extraction of some tetrahedrons or summation of tetrahedrons is not possible with this mesh. Such quantities may be extracted by using the region mesh (`mesh=reg`) with selecting the cell corresponding to the universe of the tetrahedron geometry. It should be noticed also the volume of tetrahedron is simply computed from the coordinates of the nodes and thus incorrect qunatities may be obtained when the tetrahedron geometry is clipped out through the use of nest structure of “universe”and “fill”. In such a case, the region mesh (`mesh=reg`) should be used instead.

6.2 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.

6.3 LET mesh

LET mesh begins as

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

Mesh definition is described later.

6.4 Time mesh

Time mesh is defined as

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

Mesh definition is described later.

6.5 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.

6.6 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.

In the defined mesh, emin itself is included in the minimum bin, while emax itself is not included in the maximum bin.

6.6.1 Mesh type

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

Table 6.2: 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.

Only when [t-cross] and z-type=1, nz=0 can be set with only one data. This setting provides only one tally surface of z = (*data*).

Each mesh type format is shown in followings.

6.6.2 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.

6.6.3 `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
```

6.6.4 `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
```

6.6.5 `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`= $log(M_{i+1}/M_i)$.

6.7 Other tally definitions

6.7.1 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 4.4 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`.

6.7.2 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.

6.7.3 file definition

The format to define name of output file is,

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

Note that the file name should not have the extension of '.eps' or '.vtk.' 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
```

6.7.4 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.

6.7.5 unit definition

Set output unit as

```
unit = number
```

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

6.7.6 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.

6.7.7 output definition

Set output type as

```
output = name of output
```

Details are described in each tally explanation.

6.7.8 info definition

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

```
info = 0, 1
```

6.7.9 title definition

This option is for title as

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

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

6.7.10 ANGEL parameter definition

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

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

Defined parameters are converted to the ANGEL format as

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

These parameters change the minimum of the horizontal and vertical axes, respectively. The main ANGEL parameters are shown in Table 6.3. Please see the ANGEL manual in more detail.

Table 6.3: ANGEL parameter

ANGEL parameter	Explanation
`xmin`	Minimum of horizontal axis.
`xmax`	Maximum of horizontal axis.
`ymin`	Minimum of vertical axis.
`ymax`	Maximum of vertical axis.
`xlin`	Change horizontal axis to liner scale.
`xlog`	Change horizontal axis to logarithmic scale.
`ylin`	Change vertical axis to liner scale.
`ylog`	Change vertical axis to logarithmic scale.
`cmnm`	Convert cm to nm.
`cmum`	Convert cm to μm.
`cmmm`	Convert cm to mm.
`cmmt`	Convert cm to m.
`cmkm`	Convert cm to km.

From ver. 2.89, the default size was changed from A4 to US letter. ANGEL parameters, `a4us`(US letter), `a3pp`(A3), `a4pp`(A4), `a5pp`(A5), `b3pp`(B3), `b4pp`(B4), and `b5pp`(B5) can be used to change an output size of an eps file. The characters given in parentheses represent each output size.

6.7.11 Sangel parameter definition

This definition can be used to insert all ANGEL parameters into tally output files unlike the usual ANGEL parameter definition, which can specify only ANGEL parameters defined by `p:`. For example, defining `infl:` parameter for ANGEL, the tally file includes an experimental data file, and both calculated results and the experimental data can be plotted in a figure (eps file).

The SANGEL parameter can be set using the format `sangel`=*number of definition lines*, and then writing the lines to define the ANGEL parameters. An example with `sangel` is as follows.

```
sangel = 2
infl:{exp.dat}
w: ($\theta=$ 0 deg) / X(10) Y(100)
```

where “exp.dat” is a data file of the experimental data. The data can be shown on the figure of the tally result. By using `w:` parameter, the comment “θ=0 deg” can be put at the coordinate of `X=10, Y=100` in the figure.

6.7.12 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.

6.7.13 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.

6.7.14 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.

```
        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 cannot 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.

6.7.15 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 cannot be defined in the ANGEL parameter.

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

6.7.16 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.

6.7.17 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.

6.7.18 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 is tallied.

6.7.19 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.

6.7.20 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.

6.7.21 dump definition

In the `[t-cross]`, `[t-time]`, and `[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 6.4: 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 6.5: 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 4.4), 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/u 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
```

When you use this dump parameter, `axis` and `file` are restricted to one axis and one file, and `unit` is always 1. The dumped data are written on a file named "***_dmp," where "***" indicates the file name specified by `file`="***." The normal output of the tally is written on "***." From this file, you can get the information on the total normalization factor. (In the former version of PHITS (before 2.66), the normal output was written on a configuration file (.cfg), and the dumped data were written on "***.") 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.

6.8 Function to sum up two (or more) tally results

After version 2.74, PHITS has a function to sum up two (or more) tally results. There are two methods for sum up; one is to integrate the tally results considering the history number of each simulation, while the other is to add the tally results weighted by user defined ratios. The former can be used for the parallel calculation on computers in which MPI protocol is not installed. On the other hand, the latter is suit for the simulation with different source terms whose intensities cannot be fixed before the simulation, such as that for Intensity-Modulated Radiation Therapy (IMRT).

In order to use this function, you have to satisfy the following conditions:

- `icntl` is set to 13 in [parameters] section
- The parameters for each tally results such as `mesh, axis` and `part` are identical to one another
- “sumtally” subsection is defined in the tally section that outputs one of the summing up tallies. [58] [59]

“sumtally” subsection is ignored when `icntl` parameter is not set to 13 in [parameters] section. [60] Please see the ppt slides or sample input file in `\phits\utility\sumtally\` for details.

“sumtally” subsection should be defined between the lines of `sumtally start` and `sumtally end` written in the tally section that outputs one of the summing up tallies. The `set:` definition is ignored in the sumtally subsection.

The parameters used in “sumtally” subsection are summarized in Table 6.6.

Table 6.6: Parameters used in “sumtally” subsection

name	value	description
isumtally =	1(default), 2	Summing up procedure. 1: Integration of tally results considering history number of each simulation. 2: Sum of tally results weighted by user defined ratios.
nfile = (next line)	Number filename value	Number of summing up files. Summing up file name, weighted value. Sum of the weighted values is automatically normalized to sumfactor for `isumtally=2`.
sfile =	filename	Output file name.
sumfactor =	（Omitted, D=1.0）	Normalization factor.

Users can use `isumtally=1` to manually obtain the parallel calculation results. For example, if you have one tally result “result-1.out” obtained with `maxbch=10` and `maxcas=100`, and another tally result “result-2.out” obtained with `maxbch=20` and `maxcas=100`, and if you would like to sum up the results to have same statistics as of `maxbch=30` and `maxcas=100`, you have to write:

Example 54: example for `isumtally=1`

```
sumtally start
isumtally = 1       $(D=1) sumtally option, 1:integration, 2:weighted sum
nfile = 2           $ number of tally files
   result-1.out  1.0
   result-2.out  1.0
sfile = result-s.out $ file name of output by sumtally option
sumfactor = 1.0      $ (D=1.0) normalization factor
sumtally end
```

Using this sumtally subsection, you can obtain the results for `maxcas=100` and `maxbch=30`. Please be sure that the initial random seeds for calculating “result-1.out” and “result-2.out” should be different from each other,

[58] Before ver. 2.81, this function is available even if `file` parameter is not defined in the tally section. However, after ver. 2.82, PHITS occurs the error in the setting without the definition of `file`.

[59] Before ver. 2.85, only one “sumtally” subsection can be defined in a PHITS input file.

[60] From ver. 2.88, “sumtally” can be applied to all kinds of tallies.

otherwise you would get biased results for certain random numbers. The most recommended method for changing the initial random seed is to use `irskip` parameter. The weighted value for each tally outputs are generally set to 1 for `isumtally=1`, unless you would like to change the weight of source particles. The output file obtained from this sumtally section, “result-s.out,” can be used for restart calculation by setting `istdev<0`. Note that the initial random seed in the last file of the summing up files is used.

For `isumtally=2`, the weighted summation of the tally results, $\overline{X}$, is calculated by the following equation:

$$\overline{X} = F \sum_{j=1}^{N} \frac{r_j}{r} \overline{X}_j \tag{23}$$

where F is the normalization factor defined by `sumfactor`, N is the number of summing up files defined by `nfile`, $\overline{X}_j$ is the j-th tally results, r_j is the weighted value of j-th tally, and r is the sum of r_j, i.e. $r = \sum_{j=1}^{N} r_j$. The uncertainty of the summation value, σ_X, can be calculated by

$$\sigma_X = F \sqrt{\sum_{j=1}^{N} \left(\frac{r_j}{r}\right)^2 \sigma_{X_j}^2} \tag{24}$$

where σ_{X_j} is the standard deviation of j-th tally results.

If you have two tally results “result-l.out” and “result-r.out,” and you would like to sum up them weighted by factors of 2.0 and 3.0, respectively, you have to write:

Example 55: example for `isumtally=2`

```
sumtally start
isumtally = 2       $(D=1) sumtally option, 1:integration, 2:weighted sum
nfile = 2           $ number of tally files
   result-l.out  2.0
   result-r.out  3.0
sfile = result-s.out $ file name of output by sumtally option
sumfactor = 5.0     $ (D=1.0) normalization factor
sumtally end
```

You can obtain the same results by using multi-source function, but it is more convenient to use sumtally subsection when you would like to change the weighted values for several cases. It should be mentioned that the sum of the weighted values is automatically normalized to `sumfactor` for `isumtally=2`. The output file obtained from this sumtally section, “result-s.out,” cannot be used for restart calculation.

7 Tally input format

7.1 [T-Track] section

This tally can be used to obtain the fluence in any specified region. In this tally, the track length is evaluated whenever particles pass through a specified region, as shown in Fig. 7.1, and the sum of the track lengths in units of [cm] is scored. Based on this, a particle fluence in units of [/cm^2/source] is determined by dividing the scored track lengths by the volume of the region and the number of source particles.

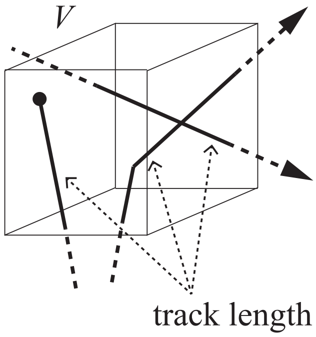


Figure 7.1: `[T-Track]` tally: track length (solid line) is calculated.

As an example, information on the detector response in a specified region can be obtained by utilizing this tally. Multiplying the fluence by the cross section (in units of cm^2) of the detector enables estimation of the number of counts in the response.

Table 7.1: `[t-track]` parameters (1)

name	value	explanation
`mesh =`	`reg, r-z, xyz, tet`	Geometry mesh: a geometry mesh subsection is required below this option.
`part =`	`all` (default), particle name	There is a maximum of six particles in a `[t-track]`.
`material =`	(omitted) `all`, number of materials	Specify materials for scoring. `all` : default (same as no definition). To set number of materials, define the material numbers in the next line. The number of materials can be set as a negative value, in which case the specified materials are not included for scoring.
(next line)	`2 5 8`	Material numbers.
`e-type =`	`1, 2, 3, 4, 5`	Energy mesh: an energy mesh subsection is required below this option.
`t-type =`	`1, 2, 3, 4, 5` (omitted)	Time mesh: a time mesh subsection is required below this option.
`a-type =`	`1, 2, -1, -2`	Angle mesh with θ degrees of freedom for `mesh=r-z`: an angle mesh subsection is required below this option.

Table 7.2: [t-track] parameters (2)

name	value	explanation
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 tet xy, yz, xz, rz t	x axis value of output data. x axis value of output data (only active with mesh=tet). 2-dimensional. Time axis.
file =	file name	Define file names. This is required by each setting of axis.
resfile =	(omitted, D=file)	Define a file name of the past tally in the restart calculation: even if several axis parameters were defined, specify only one resfile.
factor =	(omitted, D=1.0)	Normalization factor.
title =	(omitted)	Title.
angel =	(omitted)	ANGEL parameters.
sangel =	(omitted)	Special format for ANGEL parameters.
2d-type =	1, 2, 3, 4, 5, 6, 7 (omitted, D=3)	Options for 2-dimensional plot.
x-txt = y-txt = z-txt =	(omitted) (omitted) (omitted)	*x* axis title. *y* axis title. *z* axis title.
gshow =	0 (default), 1, 2, 3, 4, 5	When mesh=xyz and axis=xy,yz,xz, region border (1), material name (2), region name (3), and LAT number (4) are plotted using this option. gshow=5 outputs only material colors in pixel style when icntl=8.
rshow =	0 (default), 1, 2, 3	When mesh=xyz and axis=xy,yz,xz, region border (1), material name (2), and region name (3) are plotted using this option. A xyz mesh section must be added below this option.
ginfo =	0 (default), 1 2	No geometry check in the case of gshow or rshow> 0. Check geometry and draw a two-dimensional view with error information. Check geometry, draw two-dimensional view, and output a geometry error file (".err").

'Lethargy' in unit=3 or 13 is the natural logarithmic unit of energy defined by $\ln(E_{\rm ref}/E)$ using a reference energy $E_{\rm ref}$ and the particle energy E. Setting these units enables obtaining results in units of Lethargy, which are given as Lethargy widths, $\ln(E_{\rm high}/E_{\rm low})$, for each energy bin in the energy mesh subsection. Here $E_{\rm high}$ and $E_{\rm low}$ are the maximum and minimum values of the energy bins, respectively.

Setting unit=1,2,3,11,12 or 13 produces the mean particle fluence in a specified region calculated from the sum of the track lengths per source divided by the volume of the region. Note that, for reg mesh the volume in the [volume] section must be set; if this is not done, the particle fluence for volume=1 [cm^3], i.e., the sum of the track lengths per source, is obtained. For r-z and xyz meshes, the volume is automatically calculated. Setting unit=4 or 14 produces the sum of the track lengths per source.

θ degrees of freedom can be set in r-z mesh of this tally. The a-type mesh definition used to specify θ follows the same format as in the other angle mesh cases. The θ dependence of results can be obtained by setting axis=rad or axis=deg, where rad and deg represent θ in radians and degrees, respectively.

Table 7.3: [t-track] parameters (3)

name	value	explanation
resol =	1 (default)	This option multiplies the region line resolution by a factor of resol with the gshow or rshow option set to define the line thickness.
width =	0.5 (default)	The option defines the line thickness.
volume	(omitted)	This option defines the volume for each region for reg mesh. Volume definitions are required below this option. Values defined in [volume] are used in the case of no definition. If special description such as using ( ) is used to specify a region in reg=, its internally defined region number is output in input echo when this volume subsection is not defined.
reg vol		Volume definition. For details see Sec. 6.1.2.
iechrl =	72 (default)	Maximum number of columns for volume input echo.
volmat =	(omitted, D=9)	This option corrects the volume value for each mesh when material is defined by the xyz mesh. (0 means no correction). Value of volmat is the number of scans per one xyz mesh side.
epsout =	0 (default), 1, 2	When epsout=1, results are plotted into eps files. The eps file is named by replacing the extension with ".eps." When epsout=2, error bars are also displayed in the eps file, except for the 2-dimensional type (axis=xy, yz, xz, rz).
bmpout =	0(default), 1	Generate Bitmap figure of 2-dimensional tally output. This file is named by replacing the extension with ".bmp." When mesh=xyz and axis=xy, yz, xz, it is available.
vtkout =	0(default), 1	Output the tally results in the xyz-mesh in the input format of ParaView. This file is named by replacing the extension with ".vtk." When mesh=xyz and axis=xy, yz, xz, it is available.
vtkfmt =	0(default), 1	Format of output file for ParaView. 0: ascii, 1: binary.
foamout =	0(default), 1, 2	When foamout=1, generate a file of outputs (numerical data) with OpenFOAM field data format. This file is named by replacing the extension with ".foam." When foamout=2, generate a file of outputs (element no., center of mass *x* coordinate, *y* coordinate, *z* coordinate, output data, relative error) with CSV format. This file is named by replacing the extension with ".csv." When mesh=tet and axis=tet, it is available.
ctmin(i) =	(omitted, D=-9999)	Minimum value for *i*-th counter.
ctmax(i) =	(omitted, D= 9999)	Maximum value for *i*-th counter.
trcl =	(omitted)	Coordinate transformation number or definition for r-z or xyz mesh.
gslat =	1(default), 0	1: show lattice boundary in gshow, 0: do not show.

Table 7.4: [t-track] parameters (4)

name	value	explanation
stdcut =	(omitted, D=-1)	Threshold value of STD cut off.
multiplier =	Number of material (omitted)	Multiplier for each material: a multiplier subsection is required below this option. See Sec. 5.23.1 for its detailed usage.

When specifying stdcut, PHITS automatically stop the calculation depending on values of STD (standard deviation). This function is available when stdcut is positive and itall=0,1 is set in [parameters] section. When all relative values of STD of the tally result are smaller than stdcut at the last of one batch, PHITS finishes its calculation. If stdcut in two (or more) tally sections is set, all the results of the tally sections have to satisfy the conditions in order to work the function.

When using function to display transition of tally results and errors, mesh points to output tally results and these statistical errors can be specified by setting “anatally” subsection. This subsection should be defined between the lines of anatally start and anatally end written in the tally section. See Sec. 3.3 for more detail.

7.2 [T-Cross] section

This tally can be used to obtain the current or flux (actually, the fluence) on any specified surface. In this tally, a particle crossing the surface simply adds 1 to the current and $1/\cos\theta$ to the flux, 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 are similar but distinct physical quantities; they differ in terms of the surface element used to calculate the number of the particles crossing per unit area. The current is evaluated by dividing by the area of the surface S shown in Fig. 7.2; by contrast, the flux is calculated by dividing by $S\cos\theta$. The value of S is given in the geometry mesh subsection as `area` for `mesh=reg` and is calculated automatically for the `r-z` and `xyz` meshes.

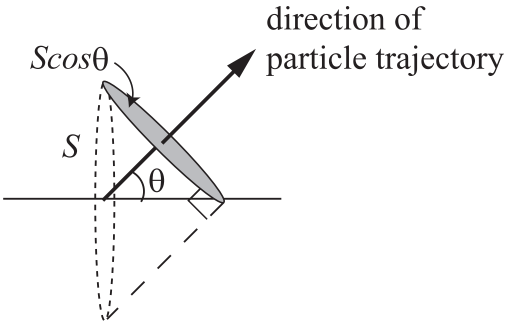


Figure 7.2: Relation between the areas S and $S\cos\theta$.

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

Table 7.5: [t-cross] parameters (1)

name	value	explanation
mesh =	reg, r-z, xyz	Geometry mesh: a geometry mesh subsection is required below this option.
part =	all (default), particle name	There is a maximum of six particles in a [t-cross].
e-type =	1, 2, 3, 4, 5	Energy mesh: an energy mesh subsection is required below this option.
t-type =	1, 2, 3, 4, 5 (omitted)	Time mesh: a time mesh subsection is required below this option.
a-type =	1, 2, -1, -2	Angle mesh (=1,2; cosine mesh, =-1,-2; angle (in units of degree) mesh): an angle mesh subsection is required below this option for output=a-curr or =oa-curr.
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]

There may be cases in which results of a tally are incorrect when r-z or xyz mesh surface agrees with that of the defined cell.

The current for a specified angle can be obtained using the angle mesh shown in Fig. 7.3. In cases where unit=4,5 6,14,15,16, the output is given as a quantity per solid angle (in steradians) calculated using the mesh size of the angle-bin defined in the angle mesh subsection.

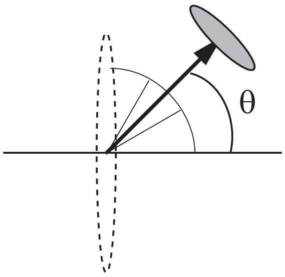


Figure 7.3: Schematic of tally using angle mesh.

'Lethargy' in unit=3,6,13,16 is the natural logarithmic unit of energy defined by $\ln(E_{\rm ref}/E)$ using a reference energy $E_{\rm ref}$ and the particle energy E. Setting these units enables obtaining results in units of Lethargy, which are given as Lethargy widths, $\ln(E_{\rm high}/E_{\rm low})$, for each energy bin in the energy mesh subsection. Here $E_{\rm high}$ and $E_{\rm low}$ are the maximum and minimum values of the energy bins, respectively.

In unit=4,5,6,14,15,16, 'sr' denotes steradians as the solid angle unit.

Table 7.6: `[t-cross]` parameters (2)

name	value	explanation

`axis =`	`eng, reg, x, y, z, r`
	`cos, the, t`
	`xy`
	`yz, xz, rz`
		x axis value of output data.
		Angle (cosine, θ[degree]) and time mesh.
		2-dimensional.
		2-dimensional, only for `enclos=1`.

`file =`	`file name`	Define file names. This is required by each setting of `axis`.
`resfile =`	(omitted, D=`file`)	Define a file name of the past tally in the restart calculation: even if several `axis` parameters were defined, specify only one `resfile`.
`factor =`	(omitted, D=1.0)	Normalization factor.
`title =`	(omitted)	Title.
`angel =`	(omitted)	ANGEL parameters.
`sangel =`	(omitted)	Special format for ANGEL parameters.
`2d-type =`	`1, 2, 3, 4, 5, 6, 7` (omitted, D=2)	Options for 2-dimensional plot.

`output =`	`flux`
	`current`
	`f-curr`
	`b-curr`
	`o-curr`
	`of-curr`
	`ob-curr`
	`a-curr`
	`oa-curr`
		Flux by surface crossing.
		Current by surface crossing.
		Forward current by surface crossing.
		Backward current by surface crossing.
		Omni[61] current by surface crossing.
		Omni[61] forward current by surface crossing.
		Omni[61] backward current by surface crossing.
		Angle mesh current by surface crossing.
		Angle mesh omni[61] current by surface crossing.

`x-txt =`
`y-txt =`
`z-txt =`
	(omitted)
	(omitted)
	(omitted)
		x axis title.
		y axis title.
		z axis title.

`gshow =`	`0` (default), `1, 2, 3, 4, 5`	When `mesh=xyz` and `axis=xy,yz,xz`, region border (1), material name (2), region name (3), and LAT number (4) are plotted using this option. `gshow=5` outputs only material colors in pixel style when `icntl=8`.
`ginfo =`	0 (default), 1 2	No geometry check in the case of `gshow` or `rshow`> 0. Check geometry and draw a two-dimensional view with error information. Check geometry, draw two-dimensional view, and output a geometry error file (".err").

`resol =`
`width =`
	1 (default)
	`0.5` (default)
		This option multiplies the region line resolution by a factor of `resol` with the `gshow` or `rshow` option set to define the line thickness.
		The option defines the line thickness.

The output options `output=f-curr, b-curr, of-curr, ob-curr` can be used in either `xyz` or `r-z` meshes. Note that in `xyz` meshes these options are available only for the *z*-direction.

[61] Omni means the energy integrated.

Table 7.7: [t-cross] parameters (3)

name	value	explanation
epsout =	0 (default), 1, 2	When epsout=1, results are plotted into eps files. The eps file is named by replacing the extension with ".eps." When epsout=2, error bars are also displayed in the eps file, except for the 2-dimensional type (axis=xy).
ctmin(i) = ctmax(i) =	(omitted, D=-9999) (omitted, D= 9999)	Minimum value for *i*-th counter. Maximum value for *i*-th counter.
trcl =	(omitted)	Coordinate transformation number or definition for r-z or xyz mesh.
dump = (next line)	Number of data Data sequence	For mesh=reg, the information is dumped on the file. If dump is negative, data are written by ascii, if positive, by binary. Define the data sequence. The history information (nocas and nobch) is necessary to use idmpmode=1.
gslat =	1(default), 0	1: show lattice boundary in gshow, 0: no.
stdcut =	(omitted, D=-1)	Threshold value of STD cut off.
multiplier =	Number of material (omitted)	Multiplier for each material.[62] Multiplier subsection is required below this parameter. See Sec. 5.23.1 for its detailed usage.
enclos =	0(default) 1	Define crossing surfaces on z and r for mesh=r-z, and define crossing surface on z for mesh=xyz. Define crossing surfaces by closed surface separated by each mesh.

In the [t-cross] tally, the dump option can only be used with reg meshes and only on axis=reg. If the dump option is set, the e-type, a-type and t-type meshes take on only the maximum and minimum values. The output option can be set as current, a-curr, or oa-curr. In using this dump parameter, axis and file are restricted to one axis and one file apiece and unit is always 1. The dumped data are written onto a file named "***_dmp", where "***" indicates the file name specified by file=***. The normal output of the tally is written on "***." From this file, information on the total normalization factor can be obtained; doing so requires setting one mesh each for e-type, a-type and t-type (in the versions of PHITS before 2.66, the normal output was written on a configuration file (.cfg) and the dumped data were written on "***"). The history information (nocas and nobch) is necessary to use idmpmode=1 for continuous calculation using the dump file; in addition, both the dump file with "_dmp" and the normal output file specified by file= are required to use idmpmode=1. The option dumpall is not compatible with this dump tally option when shared memory parallelization is active.

When specifying stdcut, PHITS automatically stop the calculation depending on values of STD (standard deviation). This function is available when stdcut is positive and itall=0,1 is set in [parameters] section. When all relative values of STD of the tally result are smaller than stdcut at the last of one batch, PHITS finishes its calculation. If stdcut in two (or more) tally sections is set, all the results of the tally sections have to satisfy the conditions in order to work the function.

From ver. 3.10, crossing surfaces are defined by closed surface separated by each mesh when enclos=1 is set with mesh=xyz or mesh=r-z. In this case, the forward direction is defined as the incoming direction, and the backward direction is defined as the outgoing direction. The total area of closed surface is used to calculate per unit area.

[62] From ver. 2.86, multiplier option can be used in [t-cross] tally.

Setting mesh=reg for the geometry mesh in this section requires defining the crossing surface by outgoing (r-from) and incoming (r-to) region number, and the area of the surface (area in units of cm^2), as shown in the example below [63].

```
        mesh = reg
         reg =  number of crossing surfaces
          r-from  r-to     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
           ...     ...     ....
           ...     ...     ....
```

In the next line of mesh=reg, give the number of crossing surfaces to tally by reg=. Furthermore, from the next line, the values of r-in r-out, and area should be written as a matrix format. The default order for this definition is r-in r-out area. The line of these column headers can be omitted. However, to change the order, rearrange and explicitly write the column headers as r-in r-out area. The skip operator non can be also used. When specifying the region number, the format (2 -5 8 9) can be used, as can the lattice and universe style (6 < 10[1 0 0] < u=3). However, any value that is not single numeric must be enclosed by ( ) .

If mesh=reg is set, the obtained current or flux is unidirectional from r-from to r-to; a bidirectional flux can be set in the third line of the above definition.

Setting the mesh=r-z defines the numbers of two crossing surface types: the number of "nz+1" crossing surfaces for z defined by $r_i - r_{i+1}$ and the number of "nr+1" crossing surfaces for r defined by $z_i - z_{i+1}$. If an r-surface coincides with the surface of the outer void, the flux on this surface is not tallied.

If mesh=xyz is set, the number of "nz+1" crossing surfaces on 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.

Setting mesh=r-z or xyz causes crossing particles to be detected in both directions on the defined surface. The forward direction is defined as the positive direction on a z surface and from the center to the exterior on an r surface. From ver. 3.05, specification of z-type=1 and nz=0 is allowed to calculate fluences of particles passing through a certain surface.

[63] Before ver. 2.96, r-in and r-out were used instead of r-from and r-to, respectively. These old parameters can be used after ver. 2.97. Note that 'in' and 'out' are reversed in this definition.

7.3 [T-Point] section

It is impractical to calculate particle fluence at a specific point or line using the [t-track] tally. To address this, the point estimator tally [t-point] was introduced to estimate these quantities within a short computational time. However, to use this tally the following conditions must be satisfied in the PHITS simulation using [t-point]:

(1) particle energy should not exceed the maximum energy of the data library used, i.e., dmax;

(2) only the fluence of neutrons and photons can be calculated by [t-point];

(3) neither event generator mode nor EGS5 should be used (e-mode=0, negs=0);

(4) the material should be uniform within a certain proximity to the point detector to avoid singularity;

(5) reflection or white boundary surface should not be used.

Please see the read-me file or the sample input file in "\phits\utility\tpoint\" for more details.

Table 7.8: [t-point] parameters (1)

name	value	explanation
point =	number of data	Option for point detectors: a subsection is required below this option.
ring =	number of data	Option for ring detectors: a subsection is required below this option.
part =	all (default), particle name	There is a maximum of six particles in a [t-point].
e-type =	1, 2, 3, 4, 5	Energy mesh: an energy mesh subsection is required below this option.
t-type =	1, 2, 3, 4, 5 (omitted)	Time mesh: a time mesh subsection is required below this option.
unit =	1, 2, 3 11, 12, 13	1: [1/cm^2/source] 2: [1/cm^2/MeV/source] 3: [1/cm^2/Lethargy/source] 11:[1/cm^2/nsec/source] 12:[1/cm^2/nsec/MeV/source] 13:[1/cm^2/Lethargy/nsec/source]
axis =	eng, t	x axis value of output data.
file =	file name	Define file names. This is required by each setting of axis.
resfile =	(omitted, D=file)	Define a file name of the past tally in the restart calculation: even if several axis parameters were defined, specify only one resfile.
factor =	(omitted, D=1.0)	Normalization factor.
title =	(omitted)	Title.
angel =	(omitted)	ANGEL parameters.
sangel =	(omitted)	Special format for ANGEL parameters.
epsout =	0 (default), 1, 2	When epsout=1, results are plotted into eps files. The eps file is named by replacing the extension with ".eps." When epsout=2, error bars are also displayed in the eps file, except for the 2-dimensional type (axis=xy, yz, xz, rz).
ctmin(i) = ctmax(i) =	(omitted, D=-9999) (omitted, D= 9999)	Minimum value for *i*-th counter. Maximum value for *i*-th counter.

'Lethargy' in unit=3 or 13 is the natural logarithmic unit of energy defined by $\ln(E_{\rm ref}/E)$ using a reference energy $E_{\rm ref}$ and the particle energy E. Setting these units enables obtaining results in units of Lethargy, which are given as Lethargy widths, $\ln(E_{\rm high}/E_{\rm low})$, for each energy bin in the energy mesh subsection. Here $E_{\rm high}$ and $E_{\rm low}$ are the maximum and minimum values of the energy bins, respectively.

In [t-point], the number of points or rings (instead of the mesh, as in other tallies) must be defined. For example, point=3 should be specified to define 3 point detectors. The maximum number of points or rings per [t-point] is 20; to set more detectors, another [t-point] tally must be defined. Point and ring detectors cannot be combined in one [t-point] tally. The information on a point or ring must be defined in t successive lines following the definition of the point or ring parameter. The point detector definition is described as follows.

```
[ T-point ]
point =    1     # number of point detectors
non     x        y       z        r0
  1     10.0     0.0     50.0     1.0
```

where x, y, and z indicate the coordinates of the point detector and r0 is the radius of the fictitious sphere (for more information on the fictitious sphere, see the read-me file in “\phits\utility\tpoint\”). These parameters are given in units of cm.

The ring detector is defined as follows:

```
[ T-point ]
ring =    1     # number of ring detectors
non   axis    ar       rr       r0
  1    z      50.0     10.0     1.0
```

where axis indicates the direction of the ring axis specified as x, y, or z, ar is the distance from the origin to the center of the ring, rr is the ring radius, and r0 is the radius of the fictitious sphere. The order of these parameters can be changed by changing the order of notation, e.g., x y z r0 can be changed to z y x r0. Aside from these factors, the parameters defined in [t-point] are the same as those in [t-track], including the multiplier option. Thus, the radiation dose at a specific point can be estimated using [t-point]. However, material, two-dimensional plot options, and transforms cannot be specified in [t-point].

Table 7.9: [t-point] parameters (2)

name	value	explanation
stdcut =	(omitted, D=-1)	Threshold value of STD cut off.
multiplier =	Number of material (omitted)	Multiplier for each material: a multiplier subsection is required below this option. See Sec. 5.23.1 for its detailed usage.

When specifying stdcut, PHITS automatically stop the calculation depending on values of STD (standard deviation). This function is available when stdcut is positive and itall=0,1 is set in [parameters] section. When all relative values of STD of the tally result are smaller than stdcut at the last of one batch, PHITS finishes its calculation. If stdcut in two (or more) tally sections is set, all the results of the tally sections have to satisfy the conditions in order to work the function.

When using function to display transition of tally results and errors, mesh points to output tally results and these statistical errors can be specified by setting “anatally” subsection. This subsection should be defined between the lines of anatally start and anatally end written in the tally section. See Sec. 3.3 for more detail.

7.4 [T-Deposit] section

This tally can be used to obtain deposition energies (heat) in certain regions. After version 3.05, it tallies the deposition energies not only by ionization of charged particles but also kerma of neutral particles. The necessity of the use of kerma approximation is automatically judged based on `[parameters]` section: neutron kerma approximation is not used when `e-mode`≥`1`, while photon kerma approximation is not used when `negs=1`. Please see Sec. 5.2.9 in more detail. Note that the neutron kerma approximation is available only for energies below `dmax(2)` (usually 20 MeV is set).

In the calculation of deposition energies by ionization of charged particle, any factor can be multiplied using the user-defined subroutines “usrdfn1.f” and “usrdfn2.f.” As examples, the default program of “usrdfn1.f” returns the dose equivalent calculated from the product of the deposit energy with the $Q(L)$ relationship defined in the ICRP60, while that of “usrdfn2.f” estimates biological dose on the basis of Microdosimetric Kinetic Model [64]. Any factor can be changed and added using this routine. Note that `letmat` should be set to water when these default programs are used (see “\ParticleTherapy\” in the recommendation setting in more detail). It should be also noted that this function cannot be applied to the deposition energy calculation by kerma approximation and track structure mode.

To tally the energy losses for particles entering the tally region, the counter should be defined by using `part` in the `[counter]` section as well as `ctmin` and `ctmax` in this tally section.

Table 7.10: `[t-deposit]` parameters (1)

name	value	explanation
`mesh =`	`reg, r-z, xyz, tet`	Geometry mesh: a geometry mesh subsection is required below this option.
`part =`	`all` (default), particle name	There is a maximum of six particles in a `[t-deposit]`. When `output=deposit` and `deposit=0`, `all` is necessary and only maximum 5 particles can be set. Deposition energies calculated by kerma approximation for neutrons and photons are scored in `part=neutron` and `photon`, respectively, as well as `part=all`.
`material =`	(omitted) `all`, number of materials	Specify materials for scoring. `all` : default (same as no definition). To set number of materials, define the material numbers in the next line. The number of materials can be set as a negative value, in which case the specified materials are not included for scoring.
(next line)	`2 5 8`	Material numbers.
`letmat =`	(omitted)	Material id for LET(dE/dx). If omitted, a real material is assumed. If a material not used in the geometry is selected, its material density must be defined in `[material]`. To calculate LET in water, define water with 1 g/cm^3 in `[material]`. When `letmat`<0 is set, PHITS automatically calculates dE/dx for water with 1 g/cm^3 for electrons and positrons: please see “ParticleTherapy” in the recommendation setting for more details.
`dedxfnc =`	(omitted, 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 by the $Q(L)$ relationship defined in the ICRP60, while that of “usrdfn2.f” estimates biological dose on the basis of Microdosimetric Kinetic Model.

[64] T.Sato *et al.*, “Biological dose estimation for charged-particle therapy using an improved PHITS code coupled with a microdosimetric kinetic model,” Radiat. Res. 171, 107-117 (2009).

Table 7.11: [t-deposit] parameters (2)

name	value	explanation
e-type =	1, 2, 3, 4, 5	Energy mesh: an energy mesh subsection is required below this option.
t-type =	1, 2, 3, 4, 5 (omitted)	Time mesh: a time mesh subsection is required below this option.
output =	dose deposit	Score the energy loss of charged particles and nuclei. Score the variance of deposition energies by individual history (an e-type subsection is required).
unit =	0, 1, 2, 3, 4, 5	0: Dose [Gy/source]; only for output=dose. 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 5: Dose [J/m^{3}1/source]; only for output=dose
deposit =	0(default), 1	Option for deposit energy plot when output=deposit. 0: distribution of total deposit energy with contribution of each particle (please use this setting normally). 1: distribution of deposit energy by each particle (default setting until ver. 2.81).
axis =	eng, reg, x, y, z, r, t tet xy, yz, xz, rz, t-eng, eng-t	x axis value of output data. x axis value of output data (only active with mesh=tet). 2-dimensional.
file =	file name	Define file names. This is required by each setting of axis.
resfile =	(omitted, D=file)	Define a file name of the past tally in the restart calculation: even if several axis parameters were defined, specify only one resfile.
factor =	(omitted, D=1.0)	Normalization factor.
title =	(omitted)	Title.
angel =	(omitted)	ANGEL parameters.
sangel =	(omitted)	Special format for ANGEL parameters.
2d-type =	1, 2, 3, 4, 5, 6, 7 (omitted, D=3)	Options for 2-dimensional plot.
x-txt = y-txt = z-txt =	(omitted) (omitted) (omitted)	*x* axis title. *y* axis title. *z* axis title.

Note that deposition energies calculated by kerma approximation are not included in this mode. It should be also noted that this mode cannot be used when weights of particles depositing energy change within a history, such as the cases that [weight window] or [importance] section is defined.

When unit is set =0 with output=dose, results can be obtained in units of [Gy/source]. When mesh=reg, the volumes of each cell should be defined in the [volume] section or set as volume parameters of the [t-deposit] section. Because absorbed dose is an intensive variable, PHITS does not output a 'sum over' in output files for unit=0. Note that, in a region including more than two materials the dose in the region does not equal the average value of the region. For example, when there are two materials with masses M_1 and M_2, and absorption energies E_1 and E_2, respectively, PHITS gives $\frac{E_1}{M_1}\frac{V_1}{V_1+V_2}+\frac{E_2}{M_2}\frac{V_1}{V_1+V_2}$ in this tally, even though its average dose is $\frac{E_1+E_2}{M_1+M_2}$. Here, V_1 and V_2 are volumes of the two materials.

By setting output=deposit, this tally can be also used for calculating the probability density of deposition energies by each history. This mode is useful for estimating the detector responses and the soft-error rates of semi-conductor devices.

When output=deposit and deposit=0, the contribution of each type of particle set in part can be obtained from the result of all. In this case, if all is not set in part, part=all is automatically set.

By setting output=deposit and deposit=1, the distribution of energies deposited by each type of particle set in part can be obtained. Thus, the sum of results obtained by setting part=*individual particle* does not equal to the distribution of energies deposited by all particles.

In the case of output=deposit and deposit=0, the statistical uncertainties of all results except for the values of part=all calculated independent of istdev as standard deviations using history variance mode.

Table 7.12: [t-deposit] parameters (3)

name	value	explanation
gshow =	0 (default), 1, 2, 3, 4, 5	When mesh=xyz and axis=xy,yz,xz, region border (1), material name (2), region name (3), and LAT number (4) are plotted using this option. gshow=5 outputs only material colors in pixel style when icntl=8.
rshow =	0 (default), 1, 2, 3	When mesh=xyz and axis=xy,yz,xz, region border (1), material name (2), and region name (3) are plotted using this option. A xyz mesh section must be added below this option.
ginfo =	0 (default), 1 2	No geometry check in the case of gshow or rshow> 0. Check geometry and draw a two-dimensional view with error information. Check geometry, draw two-dimensional view, and output a geometry error file (".err").
resol = width =	1 (default) 0.5 (default)	This option multiplies the region line resolution by a factor of resol with the gshow or rshow option set to define the line thickness. The option defines the line thickness.
volume reg vol	(omitted)	This option defines the volume for each region for reg mesh. Volume definitions are required below this option. Values defined in [volume] are used in the case of no definition. If special description such as using ( ) is used to specify a region in reg=, its internally defined region number is output in input echo when this volume subsection is not defined. Volume definition. For details see Sec. 6.1.2.
iechrl =	72 (default)	Maximum number of columns for volume input echo.
volmat =	(omitted, D=9)	This option corrects the volume value for each mesh when material is defined by the xyz mesh. (0 means no correction). Value of volmat is the number of scans per one xyz mesh side.
epsout =	0 (default), 1, 2	When epsout=1, results are plotted into eps files. The eps file is named by replacing the extension with ".eps." When epsout=2, error bars are also displayed in the eps file, except for the 2-dimensional type (axis=xy, yz, xz, rz, t-eng, eng-t).
bmpout =	0(default), 1	Generate Bitmap figure of 2-dimensional tally output. This file is named by replacing the extension with ".bmp." When mesh=xyz and axis=xy, yz, xz, it is available.
vtkout =	0(default), 1	Output the tally results in the xyz-mesh in the input format of ParaView. This file is named by replacing the extension with ".vtk." When mesh=xyz and axis=xy, yz, xz, it is available.
vtkfmt =	0(default), 1	Format of output file for ParaView. 0: ascii, 1: binary.
foamout =	0(default), 1, 2	When foamout=1, generate a file of outputs (numerical data) with OpenFOAM field data format. This file is named by replacing the extension with ".foam." When foamout=2, generate a file of outputs (element no., center of mass *x* coordinate, *y* coordinate, *z* coordinate, output data, relative error) with CSV format. This file is named by replacing the extension with ".csv." When mesh=tet and axis=tet, it is available.

Table 7.13: `[t-deposit]` parameters (4)

name	value	explanation

`ctmin(i) =`
`ctmax(i) =`
	(omitted, D=-9999)
	(omitted, D= 9999)
		Minimum value for *i*-th counter.
		Maximum value for *i*-th counter.

`trcl =`	(omitted)	Coordinate transformation number or definition for `r-z` or `xyz` mesh.
`gslat =`	1(default), 0	1: show lattice boundary in `gshow`, 0: do not show.
`dresol =`	`(omitted, D=0.0)`	Parameter for representing detector resolution. This is valid only for `output=deposit`. When `dresol`=σ_r and `dfano`=F, the deposition energy E of each event fluctuates following a Gaussian distribution with standard deviation $\sqrt{\sigma_r^2 + FE}$. If `dresol<0`, the energy resolutions (energy dependencies, spectral peak shapes, etc) are overwritten by those defined in "usresol.f."
`dfano =`	`(omitted, D=0.0)`	Parameter for representing detector resolution. Valid only for `output=deposit`. When `dresol`=σ_r and `dfano`=F, the deposition energy E of each event fluctuated following a Gaussian distribution with standard deviation $\sqrt{\sigma_r^2 + FE}$.
`stdcut =`	(omitted, D=`-1`)	Threshold value of STD cut off.
`nlatcel`	(omitted)	By setting cell number which has LAT parameter, deposit energies in cells, which have same cell number but placed in different lattice coordinates, are processed as different cells. It works only for `output=deposit`.
`nlatmem`	(omitted, D=1000)	The maximum number of cells deal with separately in one history.

Although the fano factor is generally defined as a dimensionless quantity, the `dfano` parameter is defined as a quantity with the dimension of energy.

When specifying `stdcut`, PHITS automatically stop the calculation depending on values of STD (standard deviation). This function is available when `stdcut` is positive and `itall=0,1` is set in `[parameters]` section. When all relative values of STD of the tally result are smaller than `stdcut` at the last of one batch, PHITS finishes its calculation. If `stdcut` in two (or more) tally sections is set, all the results of the tally sections have to satisfy the conditions in order to have the function work.

As an extension of sum up the deposit energies such as `reg = (100 200 300 ···)`, weighted summation option is added. Using this option in this tally, energies deposited in specified *i*-th regions for each history, $E_{\text{history},i}$ are multiplied by specified coefficients, $\alpha_i(N_{\text{list}})$, and then summed up as written by:

$$E_{\text{history}} = \sum_i \alpha_i(N_{\text{list}}) \times E_{\text{history},i}. \tag{25}$$

This option weights depending on deposited energies in each history, and it can be applied to simulation, for example soft errors in semiconductor devices. If you want to weight depending on the condition for each particle, please use user defined subroutine ("usrdfn1.f" or "usrdfn2.f") and set `dedxfnc=1 or 2`. This option can be used only `mesh=reg`, and it can be executed when you specify `reg` convolution. Formats and examples are shown below;

```
mesh = reg
  reg = weightsum
ncond = 4
   no   cell   operator   ethres   list
    1    100      ge         0.1      1
    2    100      gt         0.2      1
    3    100      eq         0.0      2
  and    200      lt         0.4      2
    4    100      le         0.5      3
ncell = 5
 cell   list01   list02   list03   list00
  100      0.0      0.5      1.0      1.0
  200      0.1      0.6      2.0      1.0
  300      0.2      0.7      3.0      1.0
  400      0.3      0.8      4.0      1.0
  500      0.4      0.9      5.0      1.0
```

First, you define the number of condition. The next line defines the order of data, i.e., condition number (`no`), cell number (`cell`), operator (`operator`), threshold energy (`ethres`), and list number (`list`). In next lines, conditions are described as many as `ncond` value. These lines means that 'list number is used when the energy deposited in the specified cell satisfies large/small relation with threshold energy.' The condition written in first is preferred when the history satisfies some conditions. If the history does not satisfy all the condition, coefficients of `list00` are used. You can set several conditions by writing `and` in no column. In the cell column, you can use the format `(  2 - 5  8 9 )` and 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 operator column, you can use greater than `gt`, greater equal `ge`, equal `eq`, less equal `le`, and less than `lt`. The unit of `ethres` is MeV.

Next, you define the number of convoluted cell. The next column defines the order of data, i.e., cell number (`cell`), and list number (`list`*xxxx*). In next lines, convoluted cell number and coefficients are described. In the cell column, you can use the format `(  2 - 5  8 9 )` and 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. It should be noted that same deposited energy is added up many times when you set same cell number many times. If you do not define `list00`, coefficients of `list00` are set to zero.

7.5 [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 the calculation of deposition energies by ionization of charged particle, In the same as `[t-deposit]`, any factor can be multiplied using the user-defined subroutines "usrdfn1.f" and "usrdfn2.f."

Table 7.14: `[t-deposit2]` parameters (1)

name	value	explanation

`mesh =`
`reg =`
	`reg`
	`r1 r2`
		Geometry mesh: a `reg` mesh subsection is required below this option.
		Specify two cell numbers.

`part =`	`all` (default), particle name	There is a maximum of six particles in a `[t-deposit2]`.

`letmat1 =`
`letmat2 =`
	(omitted)
	(omitted)
		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.
		If a material not used in the geometry is selected, its material density must be defined in `[material]`. To calculate LET in water, define water with 1 g/cm^3 in `[material]`.
		When `letmat1, or 2`<0 is set, PHITS automatically calculates dE/dx for water with 1 g/cm^3 for electrons and positrons: please see "ParticleTherapy" in the recommendation setting for more detail.

`dedxfnc1 =`
`dedxfnc2 =`
	(omitted, D=0)
	(omitted, 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 by the $Q(L)$ relationship defined in the ICRP60, while that of "usrdfn2.f" estimates biological dose on the basis of Microdosimetric Kinetic Model.

`e-type1 =`
`e-type2 =`
	`1, 2, 3, 4, 5`
	`1, 2, 3, 4, 5`
		Energy mesh: for region `r1`, an energy mesh subsection is required below this option.
		Energy mesh: for region `r2`, an energy mesh subsection is required below this option.

`t-type =`	`1, 2, 3, 4, 5` (omitted)	Time mesh: a time mesh subsection is required 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.

Table 7.15: [t-deposit2] parameters (2)

name	value	explanation
file =	file name	Define file names. This is required by each setting of axis.
resfile =	(omitted, D=file)	Define a file name of the past tally in the restart calculation: even if several axis parameters were defined, specify only one resfile.
factor =	(omitted, D=1.0)	Normalization factor.
title =	(omitted)	Title.
angel =	(omitted)	ANGEL parameters.
sangel =	(omitted)	Special format for ANGEL parameters.
2d-type =	1, 2, 3, 4, 5, 6, 7 (omitted, D=3)	Options for 2-dimensional plot.
x-txt = y-txt = z-txt =	(omitted) (omitted) (omitted)	*x* axis title. *y* axis title. *z* axis title.
volume reg vol	(omitted)	This option defines the volume for each region for reg mesh. Volume definitions are required below this option. Values defined in [volume] are used in the case of no definition. If special description such as using ( ) is used to specify a region in reg=, its internally defined region number is output in input echo when this volume subsection is not defined. Volume definition. For details see Sec. 6.1.2.
iechrl =	72 (default)	Maximum number of columns for volume input echo.
epsout =	0 (default), 1, 2	When epsout=1, results are plotted into eps files. The eps file is named by replacing the extension with ".eps." When epsout=2, error bars are also displayed in the eps file, except for the 2-dimensional type (axis=e12, e21, t-e1, t-e2, e1-t, e2-t).
ctmin(i) = ctmax(i) =	(omitted, D=-9999) (omitted, D= 9999)	Minimum value for *i*-th counter. Maximum value for *i*-th counter.
stdcut =	(omitted, D=-1)	Threshold value of STD cut off.

To tally the energy loss for each projectile particle entering the tally region, the counter should be defined using part in the [counter] section and ctmin, ctmax in this tally section.

When specifying stdcut, PHITS automatically stop the calculation depending on values of STD (standard deviation). This function is available when stdcut is positive and itall=0,1 is set in [parameters] section. When all relative values of STD of the tally result are smaller than stdcut at the last of one batch, PHITS finishes its calculation. If stdcut in two (or more) tally sections is set, all the results of the tally sections have to satisfy the conditions in order to work the function.

7.6 [T-Heat] section

This tally can be used to obtain deposition energies (heat) in certain regions using kerma approximation. However, the function to calculate deposition energy using kerma approximation was also implemented in [t-deposit] after version 3.05, and thus, we do not recommend to use [t-heat] after that version.

Although the deposition energy from neutrons is zero for e-mode≥1, it can be calculated instead from the energy loss of all charged particles and nuclei. The deposition energy from photons is also usually obtained from kerma factors. For electron=1 with electron transport, photon kerma factors are not used; instead, deposition energies are obtained from the energy loss of electrons.

Table 7.16: [t-heat] parameters (1)

name	value	explanation
mesh =	reg, r-z, xyz	Geometry mesh: a geometry mesh subsection is required 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. This is required by each setting of axis.
resfile =	(omitted, D=file)	Define a file name of the past tally in the restart calculation: even if several axis parameters were defined, specify only one resfile.
material =	(omitted) all, number of materials	Specify materials for scoring. all : default (same as no definition). When setting more than one material number, the material numbers should be specified in the next line. By setting a material number to a negative value the specified material is excluded from 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, respectively, below cut-off energy when incut>0, igcut>0, ipcut>0. leakage: Kinetic energy of particles entering the outer void. total: Total deposit energy. recoil: Kinetic energy of residual nuclei when cut-off energy emin(15-19) is set. ionization: Energy deposited by energy-loss of charged particles. low neutron: Energy deposit calculated by neutron kerma factors. photon: Energy deposit by kerma factors: (If electron=1, contributions of electrons below cut-off energy.) others: Excitation energy of residual nuclei. Setting igamma=1 makes this value 0 owing to photon emission.

Neutrons, photons, and protons below the cut-off energy are not tallied in the ncut, gcut, and pcut components but are tallied as stopped particles if, respectively, incut=0, igcut=0, or ipcut=0 in the [parameters] section. Values of incut>0, igcut>0, and ipcut>0 are tallied in the respective ncut, gcut, and pcut parts.

Table 7.17: [t-heat] parameters (2)

name	value	explanation
output = (continued)	all	In addition to the above parameters; Contributions of d, t,^{3}He, α, and residual nuclei to recoil. Contributions of p, π^+, π^-, and others to ionization (contributions of particles specified by part are output. but not plotted in the eps file). stopped particle: Kinetic energy of stopped proton, neutron, π^+, π^-, and other particles in materials (contributions of particles specified by part are output but not plotted in the eps file). others: Remaining excitation energy and fission components. 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(omitted)	When output set =all, deposit energies for ionization and stopped particles specified here are output. Results are not plotted in the eps file. Photon and neutron contributions calculated with kerma factor cannot be distinguished by specifying part. For this purpose, please see the output obtained by output=simple.
unit =	0, 1, 2	0: [Gy/source] 1: [MeV/cm^3/source] 2: [MeV/source]
factor =	(omitted, D=1.0)	Normalization factor.
title =	(omitted)	Title.
angel =	(omitted)	ANGEL parameters.
sangel =	(omitted)	Special format for ANGEL parameters.
2d-type =	1, 2, 3, 4, 5, 6, 7 (omitted, D=3)	Options for 2-dimensional plot.
x-txt = y-txt = z-txt =	(omitted) (omitted) (omitted)	x axis title. y axis title. z axis title.
gshow =	0 (default), 1, 2, 3, 4, 5	When mesh=xyz and axis=xy,yz,xz, region border (1), material name (2), region name (3), and LAT number (4) are plotted using this option. gshow=5 outputs only material colors in pixel style when icntl=8.
rshow =	0 (default), 1, 2, 3	When mesh=xyz and axis=xy,yz,xz, region border (1), material name (2), and region name (3) are plotted using this option. A xyz mesh section must be added below this option.
ginfo =	0 (default), 1 2	No geometry check in the case of gshow or rshow> 0. Check geometry and draw a two-dimensional view with error information. Check geometry, draw two-dimensional view, and output a geometry error file (".err").
resol = width =	1 (default) 0.5 (default)	This option multiplies the region line resolution by a factor of resol with the gshow or rshow option set to define the line thickness. The option defines the line thickness.

Generally speaking, heat is energy produced by the ionization of charged particles. However, in the transport simulation, transport is stopped below the set particle cut-off energy. In this case, additional components of heat, including recoil, stopped particle, and others, will be output to the heat tally. These components may change as the parameters of the transport are changed.

Table 7.18: [t-heat] parameters (3)

name	value	explanation
volume reg vol	(omitted)	This option defines the volume for each region for reg mesh. Volume definitions are required below this option. Values defined in [volume] are used in the case of no definition. If special description such as using ( ) is used to specify a region in reg=, its internally defined region number is output in input echo when this volume subsection is not defined. Volume definition. For details see Sec. 6.1.2.
iechrl =	72 (default)	Maximum number of columns for volume input echo.
volmat =	(omitted, D=9)	This option corrects the volume value for each mesh when material is defined by the xyz mesh. (0 means no correction). Value of volmat is the number of scans per one xyz mesh side.
epsout =	0 (default), 1, 2	When epsout=1, results are plotted into eps files. The eps file is named by replacing the extension with ".eps." When epsout=2, error bars are also displayed in the eps file, except for the 2-dimensional type (axis=xy, yz, xz, rz, t-eng, eng-t).
bmpout =	0(default), 1	Generate Bitmap figure of 2-dimensional tally output. This file is named by replacing the extension with ".bmp." When mesh=xyz and axis=xy, yz, xz, it is available.
vtkout =	0(default), 1	Output the tally results in the xyz-mesh in the input format of ParaView. This file is named by replacing the extension with ".vtk." When mesh=xyz and axis=xy, yz, xz, it is available.
vtkfmt =	0(default), 1	Format of output file for ParaView. 0: ascii, 1: binary.
electron =	0 (default), 1	Eelectron 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 transport are required)
ctmin(i) = ctmax(i) =	(omitted, D=-9999) (omitted, D= 9999)	Minimum value for *i*-th counter. Maximum value for *i*-th counter.
trcl =	(omitted)	Coordinate transformation number or definition for r-z or xyz mesh.
gslat =	1(default), 0	1: show lattice boundary in gshow, 0: do not show.
stdcut =	(omitted, D=-1)	Threshold value of STD cut off.

When unit is set =0 with output=dose, results can be obtained in units of [Gy/source]. When mesh=reg, the volumes of each cell should be defined in the [volume] section or set as volume parameters of the [t-deposit] section. Because absorbed dose is an intensive variable, PHITS does not output a 'sum over' in output files for unit=0. Note that, in a region including more than two materials the dose in the region does not equal the average value of the region. For example, when there are two materials with masses of M_1 and M_2, and absorption energies of E_1 and E_2, respectively, PHITS gives $\frac{E_1}{M_1}\frac{V_1}{V_1+V_2} + \frac{E_2}{M_2}\frac{V_1}{V_1+V_2}$ in this tally, even though its average dose is $\frac{E_1+E_2}{M_1+M_2}$. Here, V_1 and V_2 are volumes of the two materials.

When specifying stdcut, PHITS automatically stop the calculation depending on values of STD (standard deviation). This function is available when stdcut is positive and itall=0,1 is set in [parameters] section. When all relative values of STD of the tally result are smaller than stdcut at the last of one batch, PHITS finishes its calculation. If stdcut in two (or more) tally sections is set, all the results of the tally sections have to satisfy the conditions in order to work the function.

7.7 [T-Yield] section

This tally gives information on produced nuclei. Products produced by neutrons in the energy below `dmax(2)` are not scored in this section by default, but they are scored with `e-mode`≥1.

Table 7.19: `[t-yield]` parameters (1)

name	value	explanation
`mesh =`	`reg, r-z, xyz, tet`	Geometry mesh: a geometry mesh subsection is required below this option.
`special =`	D=0 (omitted)	When `special`> 0, nuclear reactions are repeated more than once to increase statistics.
`part =`	`all` (default), particle name	There is a maximum of six particles in a `[t-yield]`.
`material =`	(omitted) `all`, number of materials	Specify materials for scoring. `all` : default (same as no definition). To set number of materials, define the material numbers in the next line. The number of materials can be set as a negative value, in which case the specified materials are not included for scoring.
(next line)	`2 5 8`	Material numbers.
`mother =`	(omitted) `all`, number of mother nuclei	Specify mother nuclei. `all` : default (same as no definition). When setting the number of mother nuclei, define the mothers in the next line. The number of `mother` can be set as a negative value, in which case the specified mothers are not included for scoring.
(next line)	208Pb Pb	Nucleus if specified using mass: without mass, all nuclei are isotopes of Pb. To specify multiple mother groups, use multiple `[t-yield]` tallies.
`nucleus =`	(omitted) `all`, number of nuclei	Specify output nuclei. `all` : default (same as no definition). When setting number of nuclei, define the nuclei in the next line.
(next line)	208Pb Pb	Nucleus if specified with mass. Without mass, all nuclei are isotopes of Pb.
`elastic =`	1(default),0,-1	Option for contribution of recoil nucleus in elastic collision. 1: include contribution 0: exclude recoil target nucleus produced by elastic collision. -1: exclude the same nucleus as the target nucleus.
`unit =`	`1, 2`	1: [1/source] 2: [1/cm^3/source]

In the `[t-yield]` section, projectiles to tally can be specified by `part`, but the output is the sum of their contributions. To obtain each contribution separately, set multiple `[t-yield]` sections.

Table 7.20: `[t-yield]` parameters (2)

name	value	explanation

`axis =`	`reg, x, y, z, r`
	`tet`
	`xy, yz, xz, rz`
	`mass`
	`charge`
	`chart`
	`dchain`
		x axis value of output data.
		x axis value of output data (only active with `mesh=tet`).
		2-dimensional.
		Mass distribution: if the nucleus is specified, isotope distribution is used.
		Charge distribution: nucleus cannot be specified.
		Nucleus chart (x:N, y:Z). Nucleus cannot be specified.
		for DCHAIN output. All isotopes are output only for `mesh=reg` For this tally, only one axis parameter is defined in the `[t-yield]` section.

`file =`	`file name`	Define file names. This is required by each setting of `axis`.
`resfile =`	(omitted, `D=file`)	Define a file name of the past tally in the restart calculation: even if several `axis` parameters were defined, specify only one `resfile`.
`output =`	(omitted) `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 =`	(omitted, D=1.0)	Normalization factor.
`title =`	(omitted)	Title.
`angel =`	(omitted)	ANGEL parameters.
`sangel =`	(omitted)	Special format for ANGEL parameters.
`2d-type =`	`1, 2, 3, 4, 5, 6, 7` (omitted, D=3)	Options for 2-dimensional plot.

`x-txt =`
`y-txt =`
`z-txt =`
	(omitted)
	(omitted)
	(omitted)
		x axis title.
		y axis title.
		z axis title.

`gshow =`	`0` (default), `1, 2, 3, 4, 5`	When `mesh=xyz` and `axis=xy,yz,xz`, region border (1), material name (2), region name (3), and LAT number (4) are plotted using this option. `gshow=5` outputs only material colors in pixel style when `icntl=8`.
`rshow =`	`0` (default), `1, 2, 3`	When `mesh=xyz` and `axis=xy,yz,xz`, region border (1), material name (2), and region name (3) are plotted using this option. A `xyz` mesh section must be added below this option.
`ginfo =`	0 (default), 1 2	No geometry check in the case of `gshow` or `rshow`> 0. Check geometry and draw a two-dimensional view with error information. Check geometry, draw two-dimensional view, and output a geometry error file (".err").

`resol =`
`width =`
	1 (default)
	`0.5` (default)
		This option multiplies the region line resolution by a factor of `resol` with the `gshow` or `rshow` option set to define the line thickness.
		The option defines the line thickness.

`reg vol`		Volume definition. For details see Sec. 6.1.2.
`iechrl =`	`72` (default)	Maximum number of columns for volume input echo.
`volmat =`	(omitted, D=9)	This option corrects the volume value for each mesh when `material` is defined by the `xyz` mesh. (0 means no correction). Value of `volmat` is the number of scans per one `xyz` mesh side.

When `igamma=3` in the `[parameters]` section, information on the isomer production based on the EBITEM model can be obtained by setting `axis=chart` or `=dchain` in this tally.

If `output=cutoff` is specified, the parameters of `part` and `mother` are neglected.

Table 7.21: [t-yield] parameters (3)

name	value	explanation
volume reg vol	(omitted)	This option defines the volume for each region for reg mesh. Volume definitions are required below this option. Values defined in [volume] are used in the case of no definition. If special description such as using ( ) is used to specify a region in reg=, its internally defined region number is output in input echo when this volume subsection is not defined. Volume definition. For details see Sec. 6.1.2.
iechrl =	72 (default)	Maximum number of columns for volume input echo.
volmat =	(omitted, D=9)	This option corrects the volume value for each mesh when material is defined by the xyz mesh. (0 means no correction). Value of volmat is the number of scans per one xyz mesh side.
epsout =	0 (default), 1, 2	When epsout=1, results are plotted into eps files. The eps file is named by replacing the extension with ".eps." When epsout=2, error bars are also displayed in the eps file, except for the 2-dimensional type (axis=xy, yz, xz, rz).
bmpout =	0(default), 1	Generate Bitmap figure of 2-dimensional tally output. This file is named by replacing the extension with ".bmp." When mesh=xyz and axis=xy, yz, xz, it is available.
vtkout =	0(default), 1	Output the tally results in the xyz-mesh in the input format of ParaView. This file is named by replacing the extension with ".vtk." When mesh=xyz and axis=xy, yz, xz, it is available.
vtkfmt =	0(default), 1	Format of output file for ParaView. 0: ascii, 1: binary.
foamout =	0(default), 1, 2	When foamout=1, generate a file of outputs (numerical data) with OpenFOAM field data format. This file is named by replacing the extension with ".foam." When foamout=2, generate a file of outputs (element no., center of mass *x* coordinate, *y* coordinate, *z* coordinate, output data, relative error) with CSV format. This file is named by replacing the extension with ".csv." When mesh=tet and axis=tet, it is available.
ctmin(i) = ctmax(i) =	(omitted, D=-9999) (omitted, D= 9999)	Minimum value for *i*-th counter. Maximum value for *i*-th counter.
trcl =	(omitted)	Coordinate transformation number or definition for r-z or xyz mesh.
gslat =	1(default), 0	1: show lattice boundary in gshow, 0: do not show.
stdcut =	(omitted, D=-1)	Threshold value of STD cut off.

For the use of ctmin and/or ctmax in this tally, it should be noted that a progress value defined in the [counter] section will be given to the nuclear reaction products after a nuclear reaction if coll is set in the [counter] section.

When specifying stdcut, PHITS automatically stop the calculation depending on values of STD (standard deviation). This function is available when stdcut is positive and itall=0,1 is set in [parameters] section. When all relative values of STD of the tally result are smaller than stdcut at the last of one batch, PHITS finishes its calculation. If stdcut in two (or more) tally sections is set, all the results of the tally sections have to satisfy the conditions in order to work the function.

Table 7.22: `[t-yield]` parameters (4)

name	value	explanation
`ndata =`	`0`(default), `1, 2, 3`	When setting `ndata=1`, production cross section data are used in nucleon-induced reactions on α, ^{14}N, ^{16}O targets, as shown below. When setting `ndata=2`, pre-defined production cross section data in a directory specified by `file(27)` are used in nuclear reactions only on the target using the nuclear data library. This option is not available in the energy region that `e-mode` is used. When setting `ndata=3`, pre-defined production cross section data in a directory specified by `file(27)` are used in nuclear reactions on all targets. Note that except for `e-mode`, the nuclear data library and the nuclear reaction model are not used to calculate the yield on the target whose production data are defined in `file(27)`.

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

For `ndata=2,3`, a file name of the production cross section data should be given as “[element symbol]+[mass number(three digits)]+[-y-]+[incident particle(p or n)]+[.dat],” e.g., “O_016-y-n.dat” or “Pd107-y-n.dat.” Here, p and n mean proton- and neutron-induced reactions, respectively. Note that if the element has only one character in its symbol, an underline “_” must be added after the element symbol. An example of the production cross section data is as follows.

```
# ZAP = 48113  LIP = 1  INT = 2
# Elab (MeV)     sigma (b)
1.000000e-11  0.000000e+00
2.530000e-08  0.000000e+00
2.000000e+01  0.000000e+00
2.100000e+01  0.000000e+00
 ......
```

where `ZAP, LIP`, and `INT` are produced nuclei, its isomer state, and interpolation method, respectively. The produced nuclei are specified as $1000Z + A$, where Z is atomic number and A(three digits) is mass number. The isomer state is specified by `LIP`: =0; the ground state, =1; the first metastable state, =2; the second metastable state. The interpolation method `INT = (1,2,3,4,5)` are as follows.

```
INT = 1  : Histogram
      2  : Linear - Linear
      3  : Log - Linear
      4  : Linear - Log
      5  : Log - Log
```

The second line should be written as,

```
# Elab (MeV)    sigma (b)
```

The incident energy [MeV] and cross section [barn] should be written in the first and second columns from the third line. The number of the energy bin is not limited. Note that when the incident energy falls outside the energy range specified by the data, the cross section data at the minimum or maximum energies are used, respectively for the lower and higher energies. Larger nuclei than He are tallied from the data.

7.8 [T-Product] section

This tally gives particles and nuclei produced by nuclear reactions, decays, and fission, and also tallies source particles. It differs from `[t-yield]` in that the energy distribution and time distribution of produced particles and nuclei can be obtained in `[t-product]`. This tally is not available for produced particles by reactions of photons, electrons, or low energy neutrons below `dmax(2)`, which are described by data libraries. For `e-mode`≥1, however, particles and nuclei produced by neutron-induced reactions from the library can be obtained.

Table 7.23: `[t-product]` parameters (1)

name	value	explanation
`mesh =`	`reg, r-z, xyz, tet`	Geometry mesh: a geometry mesh subsection is required below this option.
`part =`	`all` (default), particle name	There is a maximum of six particles in a `[t-product]`.
`material =`	(omitted) `all`, number of materials	Specify materials for scoring. `all` : default (same as no definition). To set number of materials, define the material numbers in the next line. The number of materials can be set as a negative value, in which case the specified materials are not included for scoring.
(next line)	`2 5 8`	Material numbers.
`mother =`	(omitted) `all`, number of mother nuclei	Specify mother nuclei. `all` : default (same as no definition). When setting the number of mother nuclei, define the mothers in the next line. The number of `mother` can be set as a negative value, in which case the specified mothers are not included for scoring. When `output=atomic`, `mother` cannot be specified.
(next line)	208Pb Pb	Nucleus if specified using mass: without mass, all nuclei are isotopes of Pb. To specify multiple mother groups, use multiple `[t-product]` tallies.
`e-type =`	`1, 2, 3, 4, 5`	Energy mesh: an energy mesh subsection is required below this option.
`t-type =`	`1, 2, 3, 4, 5` (omitted)	Time mesh: a time mesh subsection is required below this option.
`a-type =`	`1, 2, -1, -2`	Angle mesh with θ degrees of freedom for `mesh=r-z`: an angle mesh subsection is required below this option.

Table 7.24: `[t-product]` parameters (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 the natural logarithmic unit of energy defined by $\ln(E_{\mathrm{ref}}/E)$ using a reference energy E_{ref} and the particle energy E. Setting these units enables obtaining results in units of Lethargy, which are given as Lethargy widths, $\ln(E_{\mathrm{high}}/E_{\mathrm{low}})$, for each energy bin in the energy mesh subsection. Here E_{high} and E_{low} are the maximum and minimum values of the energy bins, respectively.

In `unit=21 - 26` or `31 - 36`, 'sr' denotes steradians as the solid angle unit.

Table 7.25: [t-product] parameters (3)

name	value	explanation
axis =	eng, reg, x, y, z, r tet xy, yz, xz, rz t	x axis value of output data. x axis value of output data (only active with mesh=tet). 2-dimensional. Time axis.
file =	file name	Define file names. This is required by each setting of axis.
resfile =	(omitted, D=file)	Define a file name of the past tally in the restart calculation: even if several axis parameters were defined, specify only one resfile.
output =	source nuclear (default) nonela elastic decay fission atomic	Source particle. Particles from nuclear reaction, including elastic. Particles from nonelastic collision. Particles from elastic collision. Particles from decay. Particles from fission. Particles from atomic interaction.
factor =	(omitted, D=1.0)	Normalization factor.
title =	(omitted)	Title.
angel =	(omitted)	ANGEL parameters.
sangel =	(omitted)	Special format for ANGEL parameters.
2d-type =	1, 2, 3, 4, 5, 6, 7 (omitted, D=3)	Options for 2-dimensional plot.
x-txt = y-txt = z-txt =	(omitted) (omitted) (omitted)	*x* axis title. *y* axis title. *z* axis title.
gshow =	0 (default), 1, 2, 3, 4, 5	When mesh=xyz and axis=xy,yz,xz, region border (1), material name (2), region name (3), and LAT number (4) are plotted using this option. gshow=5 outputs only material colors in pixel style when icntl=8.
rshow =	0 (default), 1, 2, 3	When mesh=xyz and axis=xy,yz,xz, region border (1), material name (2), and region name (3) are plotted using this option. A xyz mesh section must be added below this option.
ginfo =	0 (default), 1 2	No geometry check in the case of gshow or rshow> 0. Check geometry and draw a two-dimensional view with error information. Check geometry, draw two-dimensional view, and output a geometry error file (".err").
resol = width =	1 (default) 0.5 (default)	This option multiplies the region line resolution by a factor of resol with the gshow or rshow option set to define the line thickness. The option defines the line thickness.
volume reg vol	(omitted)	This option defines the volume for each region for reg mesh. Volume definitions are required below this option. Values defined in [volume] are used in the case of no definition. If special description such as using ( ) is used to specify a region in reg=, its internally defined region number is output in input echo when this volume subsection is not defined. Volume definition. For details see Sec. 6.1.2.
iechrl =	72 (default)	Maximum number of columns for volume input echo.

Table 7.26: [t-product] parameters (4)

name	value	explanation
volmat =	(omitted, D=9)	This option corrects the volume value for each mesh when material is defined by the xyz mesh. (0 means no correction). Value of volmat is the number of scans per one xyz mesh side.
epsout =	0 (default), 1, 2	When epsout=1, results are plotted into eps files. The eps file is named by replacing the extension with ".eps." When epsout=2, error bars are also displayed in the eps file, except for the 2-dimensional type (axis=xy, yz, xz, rz).
bmpout =	0(default), 1	Generate Bitmap figure of 2-dimensional tally output. This file is named by replacing the extension with ".bmp." When mesh=xyz and axis=xy, yz, xz, it is available.
vtkout =	0(default), 1	Output the tally results in the xyz-mesh in the input format of ParaView. This file is named by replacing the extension with ".vtk." When mesh=xyz and axis=xy, yz, xz, it is available.
vtkfmt =	0(default), 1	Format of output file for ParaView. 0: ascii, 1: binary.
foamout =	0(default), 1, 2	When foamout=1, generate a file of outputs (numerical data) with OpenFOAM field data format. This file is named by replacing the extension with ".foam." When foamout=2, generate a file of outputs (element no., center of mass *x* coordinate, *y* coordinate, *z* coordinate, output data, relative error) with CSV format. This file is named by replacing the extension with ".csv." When mesh=tet and axis=tet, it is available.
ctmin(i) = ctmax(i) =	(omitted, D=-9999) (omitted, D= 9999)	Minimum value for *i*-th counter. Maximum value for *i*-th counter.
trcl =	(omitted)	Coordinate transformation number or definition for r-z or xyz mesh.
dump = (next line)	Number of data Data sequence	For mesh=reg, the information is dumped on the file. If dump is negative, data are written by ascii, if positive, by binary. Define the data sequence. The history information (nocas and nobch) is necessary to use idmpmode=1.
gslat =	1(default), 0	1: show lattice boundary in gshow, 0: do not show.
stdcut =	(omitted, D=-1)	Threshold value of STD cut off.

In the [t-product] tally, the dump option can only be used. If the dump option is set, the meshes of e-type and t-type have only the meaning of the maximum and minimum values. In using this dump parameter, axis and file are restricted to one axis and one file apiece and unit is always 1. The dumped data are written onto a file named "***_dmp", where "***" indicates the file name specified by file=***. The normal output of the tally is written on "***." From this file, information on the total normalization factor can be obtained; doing so requires setting one mesh each for e-type, a-type and t-type (in the versions of PHITS before 2.66, the normal output was written on a configuration file (.cfg) and the dumped data were written on "***"). The history information (nocas and nobch) is necessary to use idmpmode=1 for continuous calculation using the dump file; in addition, both the dump file with "_dmp" and the normal output file specified by file= are required to use idmpmode=1. The option dumpall is not compatible with this dump tally option when shared memory parallelization is active.

[t-product] can tally source particles, and the function can be used to modify the dump file. Dump files can be read and information written to a new dump file through modifications made by setting the dump parameter and output=source in this tally section and icntl=6 in the [parameters] section.

When specifying stdcut, PHITS automatically stop the calculation depending on values of STD (standard deviation). This function is available when stdcut is positive and itall=0,1 is set in [parameters] section. When all relative values of STD of the tally result are smaller than stdcut at the last of one batch, PHITS finishes its calculation. If stdcut in two (or more) tally sections is set, all the results of the tally sections have to satisfy the conditions in order to work the function.

For the use of ctmin and/or ctmax in this tally, it should be noted that a progress value defined in the [counter] section will be given to the nuclear reaction products after a nuclear reaction if coll is set in the [counter] section.

7.9 [T-DPA] section

This tally gives the displacement per atom (DPA) value, or number of displaced atoms per target atom, which represents radiation damage in materials irradiated by energetic particles. The result produced by this tally includes the contribution of the Coulomb scattering cross section for charged particle transportation. The DPA of low energy neutron incident reactions can be also obtained using `e-mode`≥1.

Table 7.27: `[t-dpa]` parameters (1)

name	value	explanation
`mesh =`	`reg, r-z, xyz, tet`	Geometry mesh: a geometry mesh subsection is required below this option.
`part =`	`all` (default), particle name	There is a maximum of six particles in a `[t-dpa]`.
`material =`	(omitted) `all`, number of materials	Specify materials for scoring. `all` : default (same as no definition). To set number of materials, define the material numbers in the next line. The number of materials can be set as a negative value, in which case the specified materials are not included for scoring.
(next line)	`2 5 8`	Material numbers.
`mother =`	(omitted) `all`, number of mother nuclei	Specify mother nuclei. `all` : default (same as no definition). When setting the number of mother nuclei, define the mothers in the next line. The number of `mother` can be set as a negative value, in which case the specified mothers are not included for scoring.
(next line)	208Pb Pb	Nucleus if specified using mass: without mass, all nuclei are isotopes of Pb. To specify multiple mother groups, use multiple `[t-dpa]` tallies.
`unit =`	`1, 2`	1: [DPA*1.e-24/source] 2: [DPA/source]

`axis =`	`eng, reg, x, y, z, r`
	`tet`
	`xy, yz, xz, rz`
		x axis value of output data.
		x axis value of output data (only active with `mesh=tet`).
		2-dimensional.

`file =`	`file name`	Define file names. This is required by each setting of `axis`.
`resfile =`	(omitted, `D=file`)	Define a file name of the past tally in the restart calculation: even if several `axis` parameters were defined, specify only one `resfile`.

Table 7.28: [t-dpa] parameters (2)

name	value	explanation
output =	dpa	Total DPA value.
	simple	total: total DPA value.
		cutoff1: DPA value when energies of charged particles produced by reactions are below the cutoff energy (emin).
		cutoff2: DPA value when energies of charged particles transported in materials are below the cutoff energy (emin).
		transpt: DPA value when charged particles are transported.
	all	add d, t, ^{3}He, α, and nucleus contributions as PKA to simple.
factor =	(omitted, D=1.0)	Normalization factor.
title =	(omitted)	Title.
angel =	(omitted)	ANGEL parameters.
sangel =	(omitted)	Special format for ANGEL parameters.
2d-type =	1, 2, 3, 4, 5, 6, 7 (omitted, D=3)	Options for 2-dimensional plot.
x-txt =	(omitted)	*x* axis title.
y-txt =	(omitted)	*y* axis title.
z-txt =	(omitted)	*z* axis title.
gshow =	0 (default), 1, 2, 3, 4, 5	When mesh=xyz and axis=xy,yz,xz, region border (1), material name (2), region name (3), and LAT number (4) are plotted using this option. gshow=5 outputs only material colors in pixel style when icntl=8.
rshow =	0 (default), 1, 2, 3	When mesh=xyz and axis=xy,yz,xz, region border (1), material name (2), and region name (3) are plotted using this option. A xyz mesh section must be added below this option.
ginfo =	0 (default),	No geometry check in the case of gshow or rshow> 0.
	1	Check geometry and draw a two-dimensional view with error information.
	2	Check geometry, draw two-dimensional view, and output a geometry error file (".err").
resol =	1 (default)	This option multiplies the region line resolution by a factor of resol with the gshow or rshow option set to define the line thickness.
width =	0.5 (default)	The option defines the line thickness.
volume	(omitted)	This option defines the volume for each region for reg mesh. Volume definitions are required below this option. Values defined in [volume] are used in the case of no definition. If special description such as using ( ) is used to specify a region in reg=, its internally defined region number is output in input echo when this volume subsection is not defined.
reg vol		Volume definition. For details see Sec. 6.1.2.
iechrl =	72 (default)	Maximum number of columns for volume input echo.
volmat =	(omitted, D=9)	This option corrects the volume value for each mesh when material is defined by the xyz mesh. (0 means no correction). Value of volmat is the number of scans per one xyz mesh side.

Table 7.29: [t-dpa] parameter (3)

name	value	explanation
epsout =	0 (default), 1, 2	When epsout=1, results are plotted into eps files. The eps file is named by replacing the extension with ".eps." When epsout=2, error bars are also displayed in the eps file, except for the 2-dimensional type (axis=xy, yz, xz, rz).
bmpout =	0(default), 1	Generate Bitmap figure of 2-dimensional tally output. This file is named by replacing the extension with ".bmp." When mesh=xyz and axis=xy, yz, xz, it is available.
vtkout =	0(default), 1	Output the tally results in the xyz-mesh in the input format of ParaView. This file is named by replacing the extension with ".vtk." When mesh=xyz and axis=xy, yz, xz, it is available.
vtkfmt =	0(default), 1	Format of output file for ParaView. 0: ascii, 1: binary.
foamout =	0(default), 1, 2	When foamout=1, generate a file of outputs (numerical data) with OpenFOAM field data format. This file is named by replacing the extension with ".foam." When foamout=2, generate a file of outputs (element no., center of mass *x* coordinate, *y* coordinate, *z* coordinate, output data, relative error) with CSV format. This file is named by replacing the extension with ".csv." When mesh=tet and axis=tet, it is available.
ctmin(i) = ctmax(i) =	(omitted, D=-9999) (omitted, D= 9999)	Minimum value for *i*-th counter. Maximum value for *i*-th counter.
trcl =	(omitted)	Coordinate transformation number or definition for r-z or xyz mesh.
gslat =	1(default), 0	1: show lattice boundary in gshow, 0: do not show.
stdcut =	(omitted, D=-1)	Threshold value of STD cut off.

When specifying stdcut, PHITS automatically stop the calculation depending on values of STD (standard deviation). This function is available when stdcut is positive and itall=0,1 is set in [parameters] section. When all relative values of STD of the tally result are smaller than stdcut at the last of one batch, PHITS finishes its calculation. If stdcut in two (or more) tally sections is set, all the results of the tally sections have to satisfy the conditions in order to work the function.

7.10 [T-LET] section

This tally can be used to obtain information on track length and dose as a function of the LET(dE/dx) of a given material. This tally counts the energy losses of charged particles and nuclei; accordingly, the Event Generator mode (`e-mode`≥1) must be used to transport low-energy neutrons. Note that this tally does not consider the contributions of electrons or positrons below their cutoff energies ((`emin`)) in the result.

Table 7.30: `[t-let]` parameters (1)

name	value	explanation
`mesh =`	`reg, r-z, xyz`	Geometry mesh: a geometry mesh subsection is required below this option.
`part =`	`all` (default), particle name	There is a maximum of six particles in a `[t-let]`.
`material =`	(omitted) `all`, number of materials	Specify materials for scoring. `all` : default (same as no definition). To set number of materials, define the material numbers in the next line. The number of materials can be set as a negative value, in which case the specified materials are not included for scoring.
(next line)	`2 5 8`	Material numbers.
`letmat =`	(omitted)	Material id for LET(dE/dx). If omitted, a real material is assumed. If a material not used in the geometry is selected, its material density must be defined in `[material]`. To calculate LET in water, define water with 1 g/cm^3 in `[material]`. When `letmat`<0 is set, PHITS automatically calculates dE/dx for water with 1 g/cm^3 for electrons and positrons: please see “ParticleTherapy” in the recommendation setting for more details.
`l-type =`	`1, 2, 3, 4, 5`	LET mesh: a LET mesh subsection must be added below this option. Note that the LET spectrum may have unnatural peaks when a very fine mesh, e.g., 20 meshes per order of magnitude, is set.

Table 7.31: [t-let] parameters (2)

name	value	explanation
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]
	13, 14	13: $L * f(L)$ [dimensionless], where $\int f(L)dL = 1$. 14: $L * d(L)$ [keV/μm], where $\int d(L)dL = 1$.
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. This is required by each setting of axis.
resfile =	(omitted, D=file)	Define a file name of the past tally in the restart calculation: even if several axis parameters were defined, specify only one resfile.
factor =	(omitted, D=1.0)	Normalization factor.
title =	(omitted)	Title.
angel =	(omitted)	ANGEL parameters.
sangel =	(omitted)	Special format for ANGEL parameters.
2d-type =	1, 2, 3, 4, 5, 6, 7 (omitted, D=3)	Options for 2-dimensional plot.
x-txt =	(omitted)	x axis title.
y-txt =	(omitted)	y axis title.
z-txt =	(omitted)	z axis title.

From version 3.02, new options (unit=13 and 14) were implemented. Using them, the frequency and dose probability densities of LET L, $f(L)$ and $d(L)$, respectively, can be easily calculated. The results obtained from unit=13 and 14 are proportional to those obtained from unit=2 and 4, respectively, but their absolute values are different because the integral of the probability densities are normalized to 1 for the new options. Note that unit=13 and 14 can be set only when axis=let.

Table 7.32: `[t-let]` parameters (3)

name	value	explanation
`gshow =`	`0` (default), `1, 2, 3, 4`	When `mesh=xyz` and `axis=xy,yz,xz`, region border (1), material name (2), region name (3), and LAT number (4) are plotted using this option.
`rshow =`	`0` (default), `1, 2, 3`	When `mesh=xyz` and `axis=xy,yz,xz`, region border (1), material name (2), and region name (3) are plotted using this option. A `xyz` mesh section must be added below this option.
`ginfo =`	0 (default), 1 2	No geometry check in the case of `gshow` or `rshow`> 0. Check geometry and draw a two-dimensional view with error information. Check geometry, draw two-dimensional view, and output a geometry error file (".err").
`resol =`	`1` (default)	This option multiplies the region line resolution by a factor of `resol` with the `gshow` or `rshow` option set to define the line thickness.
`width =`	`0.5` (default)	The option defines the line thickness.
`volume`	(omitted)	This option defines the volume for each region for `reg` mesh. Volume definitions are required below this option. Values defined in `[volume]` are used in the case of no definition. If special description such as using `(   )` is used to specify a region in `reg=`, its internally defined region number is output in input echo when this volume subsection is not defined.
`reg vol`		Volume definition. For details see Sec. 6.1.2.
`iechrl =`	`72` (default)	Maximum number of columns for volume input echo.
`volmat =`	(omitted, D=9)	This option corrects the volume value for each mesh when `material` is defined by the `xyz` mesh. (0 means no correction). Value of `volmat` is the number of scans per one `xyz` mesh side.
`epsout =`	`0` (default)`, 1, 2`	When `epsout=1`, results are plotted into eps files. The eps file is named by replacing the extension with ".eps." When `epsout=2`, error bars are also displayed in the eps file, except for the 2-dimensional type (`axis=xy, yz, xz, rz`).

`ctmin(i) =`
`ctmax(i) =`
	(omitted, D=-9999)
	(omitted, D= 9999)
		Minimum value for *i*-th counter.
		Maximum value for *i*-th counter.

`trcl =`	(omitted)	Coordinate transformation number or definition for `r-z` or `xyz` mesh.
`gslat =`	1(default), 0	1: show lattice boundary in `gshow`, 0: do not show.
`stdcut =`	(omitted, D=`-1`)	Threshold value of STD cut off.

When specifying `stdcut`, PHITS automatically stop the calculation depending on values of STD (standard deviation). This function is available when `stdcut` is positive and `itall=0,1` is set in `[parameters]` section. When all relative values of STD of the tally result are smaller than `stdcut` at the last of one batch, PHITS finishes its calculation. If `stdcut` in two (or more) tally sections is set, all the results of the tally sections have to satisfy the conditions in order to work the function.

7.11 [T-SED] section

Calculation of the probability density of deposition energies in microscopic sites, called the lineal energy y or specific energy z, is of great importance in the estimation of the relative biological effectiveness (RBE) of charged particles. However, such microscopic probability densities cannot be directly calculated by PHITS simulation using the [t-deposit] tally, as PHITS is designed to simulate particle motion on a 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 simulations considering productions of δ-rays and Auger electrons. Note that the name ‘SED’ derives from ‘Specific Energy Distribution.’ Details of the calculation procedure are given elsewhere. [65] [66] The function of [delta ray] section shouldn’t be used together with this tally.

Using this tally, information on the probability densities of y and z in water can be obtained. It is also possible to calculate probability densities in different materials, although the accuracy of doing so has no yet been checked. Similar to [t-let], the dose is only counted from the energy loss of charged particles and nuclei, and therefore the event generator mode (e-mode≥1) must be used to transport low-energy neutrons. The deposition energy in microscopic sites can be expressed as the deposit energy ε in MeV, the lineal energy y in keV/μm, or the specific energy z in Gy. The definitions of these quantities are given in ICRU Report 36. Usage of [t-sed] is similar to that of [t-let].

Table 7.33: [t-sed] parameters (1)

name	value	explanation
mesh =	reg, r-z, xyz	Geometry mesh: a geometry mesh subsection is required below this option.
part =	all (default), particle name	There is a maximum of six particles in a [t-sed].
material =	(omitted) all, number of materials	Specify materials for scoring. all : default (same as no definition). To set number of materials, define the material numbers in the next line. The number of materials can be set as a negative value, in which case the specified materials are not included for scoring.
(next line)	2 5 8	Material numbers.
letmat =	(omitted)	Material id for LET(dE/dx). If omitted, a real material is assumed. If a material not used in the geometry is selected, its material density must be defined in [material]. To calculate LET in water, define water with 1 g/cm^3 in [material]. When letmat<0 is set, PHITS automatically calculates dE/dx for water with 1 g/cm^3 for electrons and positrons: please see “ParticleTherapy” in the recommendation setting for more details.
se-unit =	1, 2, 3	Unit of deposition energy in microscopic sites. 1: deposit energy ε in MeV 2: lineal energy y in keV/μm 3: specific energy z in Gy
cdiam =	(omitted, D=1.0)	Diameter of the microscopic site in μm. A value from 0.001 to 2.0 can be selected.

[65] 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).

[66] 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).

Table 7.34: `[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`): an energy mesh subsection (specified in `ne, emin, emax`, etc.) is required below this option. If a 'Warning: Z bin is not enough!' message occurs, the `emin, emax`, and `ne` parameters must be changed. This warning indicates that the microdosimetric function cannot calculate the y (or z) distribution because the range is too small or the mesh resolution is too poor. For example, set `se-type=3, emin=0.01, emax=10000.0, ne=60` or higher for calculating the y distribution for cite diameter = 1 μm (`cdiam=1.0`, `se-unit=2`).
`unit =`	`1, 2, 3, 4, 5, 6` 7,8	1: Track [cm/(keV/μm)/source] 2: Dose [MeV/(keV/μm)/source], proportional to $y * f(y)$. 3: Track [cm/ln(keV/μm)/source] 4: Dose [MeV/ln(keV/μm)/source], proportional to $y * d(y)$. 5: Track [cm/source] 6: Dose [MeV/source] 7: $y * f(y)$ [dimensionless], where $\int f(y)dy = 1$. 8: $y * d(y)$ [keV/μm], where $\int d(y)dy = 1$. These 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. This is required by each setting of `axis`.
`resfile =`	(omitted, D=`file`)	Define a file name of the past tally in the restart calculation: even if several `axis` parameters were defined, specify only one `resfile`.
`factor =`	(omitted, D=1.0)	Normalization factor.
`title =`	(omitted)	Title.
`angel =`	(omitted)	ANGEL parameters.
`sangel =`	(omitted)	Special format for ANGEL parameters.
`2d-type =`	`1, 2, 3, 4, 5, 6, 7` (omitted, D=3)	Options for 2-dimensional plot.

`x-txt =`
`y-txt =`
`z-txt =`
	(omitted)
	(omitted)
	(omitted)
		x axis title.
		y axis title.
		z axis title.

From version 3.02, new options (`unit=7` and `8`) were implemented. Using them, the frequency and dose probability densities of y, $f(y)$ and $d(y)$, respectively, can be easily calculated. The results obtained from `unit=7` and `8` are proportional to those obtained from `unit=2` and `4`, respectively, but their absolute values are different because the integral of the probability densities are normalized to 1 in the cases of the new options. Note that `unit=13` and `14` can be set only when `axis=sed`.

Table 7.35: [t-sed] parameters (3)

name	value	explanation
gshow =	0 (default), 1, 2, 3, 4, 5	When mesh=xyz and axis=xy,yz,xz, region border (1), material name (2), region name (3), and LAT number (4) are plotted using this option. gshow=5 outputs only material colors in pixel style when icntl=8.
rshow =	0 (default), 1, 2, 3	When mesh=xyz and axis=xy,yz,xz, region border (1), material name (2), and region name (3) are plotted using this option. A xyz mesh section must be added below this option.
ginfo =	0 (default), 1 2	No geometry check in the case of gshow or rshow> 0. Check geometry and draw a two-dimensional view with error information. Check geometry, draw two-dimensional view, and output a geometry error file (".err").
resol = width =	1 (default) 0.5 (default)	This option multiplies the region line resolution by a factor of resol with the gshow or rshow option set to define the line thickness. The option defines the line thickness.
volume reg vol	(omitted)	This option defines the volume for each region for reg mesh. Volume definitions are required below this option. Default values are given in input echo in the case of no definition. Volume definition. For details see Sec. 6.1.2.
iechrl =	72 (default)	Maximum number of columns for volume input echo.
volmat =	(omitted, D=9)	This option corrects the volume value for each mesh when material is defined by the xyz mesh. (0 means no correction). Value of volmat is the number of scans per one xyz mesh side.
epsout =	0 (default), 1, 2	When epsout=1, results are plotted into eps files. The eps file is named by replacing the extension with ".eps." When epsout=2, error bars are also displayed in the eps file, except for the 2-dimensional type (axis=xy, yz, xz, rz).
ctmin(i) = ctmax(i) =	(omitted, D=-9999) (omitted, D= 9999)	Minimum value for *i*-th counter. Maximum value for *i*-th counter.
trcl =	(omitted)	Coordinate transformation number or definition for r-z or xyz mesh.
gslat =	1(default), 0	1: show lattice boundary in gshow, 0: do not show.
stdcut =	(omitted, D=-1)	Threshold value of STD cut off.

When specifying stdcut, PHITS automatically stop the calculation depending on values of STD (standard deviation). This function is available when stdcut is positive and itall=0,1 is set in [parameters] section. When all relative values of STD of the tally result are smaller than stdcut at the last of one batch, PHITS finishes its calculation. If stdcut in two (or more) tally sections is set, all the results of the tally sections have to satisfy the conditions in order to work the function.

7.12 [T-Time] section

This tally gives information on the number of particles of energy cut-off, escape, and decay in the time mesh [nsec]. Energy spectra of the particles can be also obtained. Especially, this is the unique tally can obtain energy spectra of only the particles of energy cut-off.

Table 7.36: `[t-time]` parameters (1)

name	value	explanation
`mesh =`	`reg, r-z, xyz`	Geometry mesh: a geometry mesh subsection is required below this option.
`part =`	`all` (default), particle name	There is a maximum of six particles in a `[t-time]`.
`material =`	(omitted) `all`, number of materials	Specify materials for scoring. `all` : default (same as no definition). To set number of materials, define the material numbers in the next line. The number of materials can be set as a negative value, in which case the specified materials are not included for scoring.
(next line)	`2 5 8`	Material numbers.
`t-type =`	`1, 2, 3, 4, 5`	Time mesh: a time mesh subsection is required below this option.
`e-type =`	`1, 2, 3, 4, 5`	Energy mesh: an energy mesh subsection is required 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 =`	`t, eng, reg, x, y, z, r`
	`xy, yz, xz, rz`
		x axis value of output data.
		2-dimensional.

`file =`	`file name`	Define file names. This is required by each setting of `axis`.
`resfile =`	(omitted, D=`file`)	Define a file name of the past tally in the restart calculation: even if several `axis` parameters were defined, specify only one `resfile`.

`output =`	`all`
	`cutoff`
	`escape`
	`decay`
		Energy cut off, escape, and decay particles.
		Energy cut off particles.
		Escape particles.
		Eecay particles.

`factor =`	(omitted, D=1.0)	Normalization factor.
`title =`	(omitted)	Title.
`angel =`	(omitted)	ANGEL parameters.
`sangel =`	(omitted)	Special format for ANGEL parameters.
`2d-type =`	`1, 2, 3, 4, 5, 6, 7` (omitted, D=3)	Options for 2-dimensional plot.

`x-txt =`
`y-txt =`
`z-txt =`
	(omitted)
	(omitted)
	(omitted)
		x axis title.
		y axis title.
		z axis title.

Table 7.37: [t-time] parameters (2)

name	value	explanation
gshow =	0 (default), 1, 2, 3, 4, 5	When mesh=xyz and axis=xy,yz,xz, region border (1), material name (2), region name (3), and LAT number (4) are plotted using this option. gshow=5 outputs only material colors in pixel style when icntl=8.
rshow =	0 (default), 1, 2, 3	When mesh=xyz and axis=xy,yz,xz, region border (1), material name (2), and region name (3) are plotted using this option. A xyz mesh section must be added below this option.
ginfo =	0 (default), 1 2	No geometry check in the case of gshow or rshow> 0. Check geometry and draw a two-dimensional view with error information. Check geometry, draw two-dimensional view, and output a geometry error file (".err").
resol = width =	1 (default) 0.5 (default)	This option multiplies the region line resolution by a factor of resol with the gshow or rshow option set to define the line thickness. The option defines the line thickness.
volume reg vol	(omitted)	This option defines the volume for each region for reg mesh. Volume definitions are required below this option. Values defined in [volume] are used in the case of no definition. If special description such as using ( ) is used to specify a region in reg=, its internally defined region number is output in input echo when this volume subsection is not defined. Volume definition. For details see Sec. 6.1.2.
iechrl =	72 (default)	Maximum number of columns for volume input echo.
volmat =	(omitted, D=9)	This option corrects the volume value for each mesh when material is defined by the xyz mesh. (0 means no correction). Value of volmat is the number of scans per one xyz mesh side.
epsout =	0 (default), 1, 2	When epsout=1, results are plotted into eps files. The eps file is named by replacing the extension with ".eps." When epsout=2, error bars are also displayed in the eps file, except for the 2-dimensional type (axis=xy, yz, xz, rz).
bmpout =	0(default), 1	Generate Bitmap figure of 2-dimensional tally output. This file is named by replacing the extension with ".bmp." When mesh=xyz and axis=xy, yz, xz, it is available.
vtkout =	0(default), 1	Output the tally results in the xyz-mesh in the input format of ParaView. This file is named by replacing the extension with ".vtk." When mesh=xyz and axis=xy, yz, xz, it is available.
vtkfmt =	0(default), 1	Format of output file for ParaView. 0: ascii, 1: binary.
ctmin(i) = ctmax(i) =	(omitted, D=-9999) (omitted, D= 9999)	Minimum value for *i*-th counter. Maximum value for *i*-th counter.
trcl =	(omitted)	Coordinate transformation number or definition for r-z or xyz mesh.

Table 7.38: [t-time] parameters (3)

name	value	explanation
dump =	Number of data	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. The history information (nocas and nobch) is necessary to use idmpmode=1.
gslat =	1(default), 0	1: show lattice boundary in gshow, 0: no.
stdcut =	(omitted, D=-1)	Threshold value of STD cut off.

In the [t-time] tally, the dump option can only be used only with output=cutoff. If the dump option is set, the e-type and t-type meshes take on only the maximum and minimum values. In using this dump parameter, axis and file are restricted to one axis and one file apiece and unit is always 1. The dumped data are written onto a file named "***_dmp," where "***" indicates the file name specified by file=***. The normal output of the tally is written on "***." From this file, information on the total normalization factor can be obtained; doing so requires setting one mesh each for e-type and t-type (in the versions of PHITS before 2.66, the normal output was written on a configuration file (.cfg) and the dumped data were written on "***"). The history information (nocas and nobch) is necessary to use idmpmode=1 for continuous calculation using the dump file; in addition, both the dump file with "_dmp" and the normal output file specified by file= are required to use idmpmode=1. The option dumpall is not compatible with this dump tally option when shared memory parallelization is active.

By the dump option, similar files to ncut, gcut and pcut files can be created for the sequential calculations of another transport code.

When specifying stdcut, PHITS automatically stop the calculation depending on values of STD (standard deviation). This function is available when stdcut is positive and itall=0,1 is set in [parameters] section. When all relative values of STD of the tally result are smaller than stdcut at the last of one batch, PHITS finishes its calculation. If stdcut in two (or more) tally sections is set, all the results of the tally sections have to satisfy the conditions in order to work the function.

7.13 [T-Interact] section (formerly named [T-Star])

This tally gives the number of interactions occurred in the specified regions. It was formerly named [t-star] because it can calculate the star density, but we changed the name from version 3.04 in order to explicitly state that it is the tally for counting the number of interactions. Note that the former name, [t-star], can still be accepted even after version 3.04. To calculate the number of atomic interactions using this tally, use the EGS5 mode (negs=1).

Table 7.39: [t-interact] parameters (1)

name	value	explanation
mesh =	reg, r-z, xyz	Geometry mesh: a geometry mesh subsection is required below this option.
MorP =	mean, prob(omissible)	Output the mean number (mean) or probability density (prob) of interactions. Only when mesh=reg, select prob.
part =	all (default), particle name	There is a maximum of six particles in a [t-interact]. Specify projectile particles of the reaction.
material =	(omitted) all, number of materials	Specify materials for scoring. all : default (same as no definition). To set number of materials, define the material numbers in the next line. The number of materials can be set as a negative value, in which case the specified materials are not included for scoring.
(next line)	2 5 8	Material numbers.
mother =	(omitted) all, number of mother nuclei	Specify mother nuclei. all : default (same as no definition). When setting the number of mother nuclei, define the mothers in the next line. The number of mother can be set as a negative value, in which case the specified mothers are not included for scoring. When output=atomic, mother cannot be specified.
(next line)	208Pb Pb	Nucleus if specified using mass: without mass, all nuclei are isotopes of Pb. To specify multiple mother groups, use multiple [t-interact] tallies.
nucleus =	(omitted) all, number of nuclei	Specify output nuclei. all : default (same as no definition). When setting number of nuclei, define the nuclei in the next line.
(next line)	208Pb Pb	Nucleus if specified with mass. Without mass, all nuclei are isotopes of Pb.
e-type =	1, 2, 3, 4, 5	Energy mesh: an energy mesh subsection is required below this option.
t-type =	1, 2, 3, 4, 5 (omitted)	Time mesh: a time mesh subsection is required below this option.
unit =	1, 2 11, 12	1: [1/cm^3/source] 2: [1/cm^3/MeV/source] 11:[1/cm^3/nsec/source] 12:[1/cm^3/nsec/MeV/source]
axis =	eng, reg, x, y, z, r xy, yz, xz, rz t act	x axis value of output data. 2-dimensional. Time axis. Number of interaction (for MorP=prob).
maxact =	(omitted, D=100)	Maximum number of interactions per history for axis=act. If the number of interactions exceeds this value, the history is counted as ‘overflow,’ whose probability is outputted just above ‘sum over’ line.

Table 7.40: `[t-interact]` parameters (2)

name	value	explanation
`file =`	`file name`	Define file names. This is required by each setting of `axis`.
`resfile =`	(omitted, D=`file`)	Define a file name of the past tally in the restart calculation: even if several `axis` parameters were defined, specify only one `resfile`.
`output =`		Reaction type. Only one can be set.
	`all`	All reactions.
	`decay`	Decay reaction.
	`elastic`	Elastic scattering.
	`nuclear`	Nuclear reaction excluding elastic scattering. This corresponds to `nonela` in `[t-product]`.
	`fission`	Fission.
	`absorption`	Absorption.
	`heavyion`	Heavy Ion reaction.
	`transmut`	Reactions inducing transmutation of target nuclei.
	`atomic`	Atomic interaction excluding multiple scattering.
	`deltaray`	Delta-ray production.
	`knockelec`	Production of knock-on electrons by electrons and positrons.
	`atmflu`	Atomic fluorescence x-ray emission.
	`auger`	Auger electron emission.
	`brems`	Bremsstrahlung.
	`photoelec`	Photoelectric effect.
	`compton`	Compton scattering.
	`pairprod`	Electron-positron pair production.
	`annih`	Positron annihilation.
	`ets_elast`	Elastic scattering in the track-structure mode.
	`ets_ioniz`	Ionization in track-structure mode.
	`ets_e-exc`	Electronic excitation in track-structure mode.
	`ets_v-exc`	Vibration excitation in track-structure mode.
	`ets_p-exc`	Photon excitation in track-structure mode.
	`ets_r-exc`	Rotation excitation in track-structure mode.
	`ets_dea`	Dissociative electron attachment in track-structure mode.
	`ets_ioniz_e-exc`	Sum of the ionization and electronic excitation in track-structure mode.
`factor =`	(omitted, D=1.0)	Normalization factor.
`title =`	(omitted)	Title.
`angel =`	(omitted)	ANGEL parameters.
`sangel =`	(omitted)	Special format for ANGEL parameters.
`2d-type =`	`1, 2, 3, 4, 5, 6, 7` (omitted, D=3)	Options for 2-dimensional plot.
`x-txt =`	(omitted)	*x* axis title.
`y-txt =`	(omitted)	*y* axis title.
`z-txt =`	(omitted)	*z* axis title.
`gshow =`	`0` (default), `1, 2, 3, 4, 5`	When `mesh=xyz` and `axis=xy,yz,xz`, region border (1), material name (2), region name (3), and LAT number (4) are plotted using this option. `gshow=5` outputs only material colors in pixel style when `icntl=8`.
`rshow =`	`0` (default), `1, 2, 3`	When `mesh=xyz` and `axis=xy,yz,xz`, region border (1), material name (2), and region name (3) are plotted using this option. A `xyz` mesh section must be added below this option.

Table 7.41: [t-interact] parameters (3)

name	value	explanation
ginfo =	0 (default), 1 2	No geometry check in the case of gshow or rshow> 0. Check geometry and draw a two-dimensional view with error information. Check geometry, draw two-dimensional view, and output a geometry error file (".err").
resol = width =	1 (default) 0.5 (default)	This option multiplies the region line resolution by a factor of resol with the gshow or rshow option set to define the line thickness. The option defines the line thickness.
volume reg vol	(omitted)	This option defines the volume for each region for reg mesh. Volume definitions are required below this option. Values defined in [volume] are used in the case of no definition. If special description such as using ( ) is used to specify a region in reg=, its internally defined region number is output in input echo when this volume subsection is not defined. Volume definition. For details see Sec. 6.1.2.
iechrl =	72 (default)	Maximum number of columns for volume input echo.
volmat =	(omitted, D=9)	This option corrects the volume value for each mesh when material is defined by the xyz mesh. (0 means no correction). Value of volmat is the number of scans per one xyz mesh side.
epsout =	0 (default), 1, 2	When epsout=1, results are plotted into eps files. The eps file is named by replacing the extension with ".eps." When epsout=2, error bars are also displayed in the eps file, except for the 2-dimensional type (axis=xy, yz, xz, rz).
bmpout =	0(default), 1	Generate Bitmap figure of 2-dimensional tally output. This file is named by replacing the extension with ".bmp." When mesh=xyz and axis=xy, yz, xz, it is available.
vtkout =	0(default), 1	Output the tally results in the xyz-mesh in the input format of ParaView. This file is named by replacing the extension with ".vtk." When mesh=xyz and axis=xy, yz, xz, it is available.
vtkfmt =	0(default), 1	Format of output file for ParaView. 0: ascii, 1: binary.
ctmin(i) = ctmax(i) =	(omitted, D=-9999) (omitted, D= 9999)	Minimum value for *i*-th counter. Maximum value for *i*-th counter.
trcl =	(omitted)	Coordinate transformation number or definition for r-z or xyz mesh.
gslat =	1(default), 0	1: show lattice boundary in gshow, 0: do not show.
stdcut =	(omitted, D=-1)	Threshold value of STD cut off.

When specifying stdcut, PHITS automatically stop the calculation depending on values of STD (standard deviation). This function is available when stdcut is positive and itall=0,1 is set in [parameters] section. When all relative values of STD of the tally result are smaller than stdcut at the last of one batch, PHITS finishes its calculation. If stdcut in two (or more) tally sections is set, all the results of the tally sections have to satisfy the conditions in order to work the function.

7.14 [T-Dchain] section

This tally is used for generating input files for DCHAIN. Figure 7.4 illustrates the flowchart of the connection calculation between PHITS and DCHAIN. Please see the read-me file in "\phits\recommendation\dchain\" folder for more details of the procedure for the connection. When publishing results obtained using this tally, please provide a reference to the document [67] in footnote below.

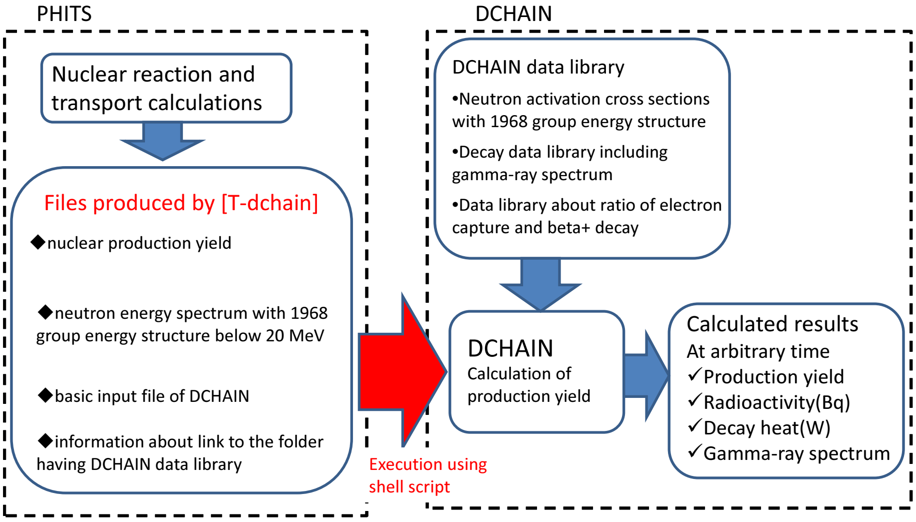

PHITS
Nuclear reaction and transport calculations
Files produced by [T-dchain]
◆nuclear production yield
◆neutron energy spectrum with 1968 group energy structure below 20 MeV
◆basic input file of DCHAIN
◆information about link to the folder having DCHAIN data library
Execution using shell script
DCHAIN
DCHAIN data library
•Neutron activation cross sections with 1968 group energy structure
•Decay data library including gamma-ray spectrum
•Data library about ratio of electron capture and beta+ decay
DCHAIN
Calculation of production yield
Calculated results
At arbitrary time
✓Production yield
✓Radioactivity(Bq)
✓Decay heat(W)
✓Gamma-ray spectrum

Figure 7.4: Concept of the connection calculation between PHITS and DCHAIN.

In the PHITS calculation, `[t-dchain]` automatically creates `[t-track]` and `[t-yield]` as well as the input file of DCHAIN. The `[t-track]` tally calculates the neutron energy spectra below 20 MeV with a 1968-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 calculation, the neutron energy spectra are multiplied by the activation cross section contained in the DCHAIN 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 evaluates radioactivity, nuclide, decay heat, and the gamma energy spectrum at irradiation and cooling time.

Note that the time variation in particle transport simulation or that given by time distribution defined in the `[source]` section is independent of the time variation in the DCHAIN calculation.

Setting `e-mode`≥1 in the `[parameters]` section enables calculation of the yields of radioactive nuclides produced by low-energy neutron reactions below 20 MeV using PHITS instead of the activation cross sections contained in the DCHAIN data library. However, the accuracy of the event generator mode relative to that of the DCHAIN data library in terms of calculating the residual-nuclide yields has not been verified, and it is therefore recommended that the use set `e-mode=0` (default) in the PHITS calculation using `[t-dchain]`. Note that activations from the originally activated target are not included in the DCHAIN calculation.

From ver. 3.00, the natural isotope expansion defined in `[material]` was effective in input files of DCHAIN generated by `[t-dchain]`. Note that in the case that one nucleus is defined two (or more) times in a material, only the latter one is effective. For example, if a material is defined as follows:

```
    MAT[1] Fe 1  56Fe 1
```

a contribution of `56Fe` included in `Fe` is ignored in the DCHAIN calculation.

[67] Tetsuya Kai, et al., "DCHAIN-SP 2001: High Energy Particle Induced Radioactivity Calculation Code", JAERI-Data/Code-2001-016 (2001) in Japanese

Table 7.42: [t-dchain] parameters (1)

name	value	explanation
mesh=	reg, xyz	Geometry mesh (Currently ONLY reg mesh and xyz mesh can be specified): a geometry mesh subsection must be added below this option. (reg = cell number).
file=	file name	Input file name of DCHAIN. Any extension except .dtrk, .dyld, or .dout can be used.
title=	(omitted)	Title.
stdcut=	(omitted, D=-1)	Threshold value of STD cut off.
ndata =	0(default), 1, 2, 3	When setting ndata=1, production cross section data are used in nucleon-induced reactions on α, ^{14}N, ^{16}O targets, as shown below. When setting ndata=2, pre-defined production cross section data in a directory specified by file(27) are used in nuclear reactions only on the target using the nuclear data library. This option is not available in the energy region that e-mode is used. When setting ndata=3, pre-defined production cross section data in a directory specified by file(27) are used in nuclear reactions on all targets. Note that except for e-mode, the nuclear data library and the nuclear reaction model are not used to calculate the yield on the target whose production data are defined in file(27).
timeevo= (next line)	number time and factor	Number of irradiation and cooling steps in DCHAIN calculation. time: time step of irradiation and/or cooling. factor: normalized factor for beam intensity. Time should be calculated from the end of the last step and 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 unit. *See example of input for [t-dchain] tally in Example 56.
outtime= (next line)	number time	Number of output timings in the DCHAIN calculation. Output timing. If a positive value is given as the output timing, it is calculated from the start of the first irradiation step. If a negative value is given, it is calculated from the end of last irradiation step. Except for this positive and negative rule, the format for specifying the timing is the same as that for timeevo. The timing cannot be specified until after all steps defined in timeevo are finished. To output the timing when no radioactive nuclide exists, e.g., 0.0 m, set iprtb2=0 in the [t-dchain] section.

When specifying stdcut, PHITS automatically stop the calculation depending on values of STD (standard deviation). This function is available when stdcut is positive and itall=0,1 is set in [parameters] section. When all relative values of STD of the tally result are smaller than stdcut at the last of one batch, PHITS finishes its calculation. If stdcut in two (or more) tally sections is set, all the results of the tally sections have to satisfy the conditions in order to work the function.

See Sec. 7.7 for ndata.

Table 7.43: [t-dchain] parameters (2)

name	value	explanation
amp=	(omitted,D=1.0)	Power of source (source/second).
target =	0, 1 (omitted,D=0)	0: OFF; Not necessary to write target information: the information is automatically determined from the [material], [cell], and [volume] sections. 1: ON; Necessary to write target information. *To add nuclides that are not defined in the [material] in the DCHAIN calculation, and/or the volume is not set in the [volume], write target=1 and provide the related information as in Example 57. *See example of target subsection for target=1 in Example 57.
iertdcho =	1 (default), 0	Toggles propagation of PHITS's [t-yield] statistical uncertainties (found in the "***_err.dyld" output file) through the DCHAIN decay and buildup calculations. When iertdcho=1, the uncertainties are propagated through the DCHAIN calculation and reported in extra columns in the primary activation output file ("spd-act.out" / "***.act"). When iertdcho=0, the uncertainties are not propagated and the extra columns for the uncertainties are omitted, leaving the file with the same formatting from before the implementation of this functionality.
inxslib =	(D=2)	Options for selection of neutron reaction cross section library to be used by DCHAIN.
	0	JEFF-3.1A (DCHAIN-SP's default from 2009-2019)
	1	FENDL/A-2.0 (DCHAIN-SP's default from 2001-2009)
	2 (default)	JENDL/AD-2017
idcylib =	(D=5)	Options for selection of decay data library to be used by DCHAIN.
	0	A hybrid library composed of evaluations and revisions from EAF-3.1, FENDL/D-1, ENSDF (1997), Table of Isotopes 8th ed. (1996), and Chart of the Nuclides 1996. (DCHAIN-SP's only decay library from 2001-2019)
	2	JENDL/DDF-2015
	3	ENDF/B-VIII.0
	4	Hybrid library of ENDF/B-VIII.0 where available and JENDL/DDF-2015 for nuclides exclusive to that library.
	5	Hybrid library of JENDL/DDF-2015 where available and ENDF/B-VIII.0 for nuclides exclusive to that library.

In addition to the above parameters, the DCHAIN parameters can also be specified in [t-dchain] section. The specifiable parameters are:

> imode, jmode, idivs, iregon, inmtcf, ichain, itdecs, itdecn, isomtr, ifisyd, ifisye, iyild, iggrp, ibetap, acmin, istabl, igsdef, igsorg, ebeam, prodnp, hnxslib, hdcylib, iwrtchn, chrlvth, iwrchdt, iwrchss

The respective meanings of these parameters are given in the DCHAIN manual, where you can find in the "\phits\dchain-sp\manual\" folder.

Example 56: Example of input for `[t-dchain]` tally

```
   mesh = reg                     <-region mesh
    reg = 100                     <-cell number
   file = testDC.spd              <-file name of DCHAIN input
  title = [t-dchain] test calc.
    amp = 1.0E12                  <-source power (source/sec)

timeevo = 4                       <-number of irradiation and cooling steps
         3.0 h  1.0               <-irradiation for 3 hours
         2.0 h  0.0               <-cooling for 2 hours
         3.5 h  1.0               <-irradiation for 3.5 hours
        15.5 h  0.0               <-cooling for 15.5 hours

outtime = 3                       <-number of output timing
         3.0 h                    <-3 hours later from the 1st irradiation start time
        -1.0 h                    <-1 hour later from the end of the last irradiation step
        -3.0 h                    <-3 hour later from the end of the last irradiation step
```

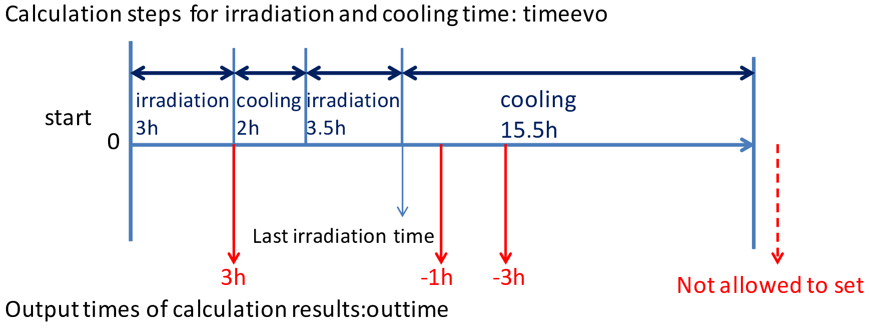

Calculation steps for irradiation and cooling time: timeevo
start
0
irradiation
3h
cooling
2h
irradiation
3.5h
cooling
15.5h
Last irradiation time
3h
-1h
-3h
Not allowed to set
Output times of calculation results:outtime

Figure 7.5: Relation between steps for irradiation and cooling and output times.

Example 57: 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
```

To indicate an isotope, the symbol of the chemical element must be connected with its atomic number using the character '-.'

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 placement of the data folder for DCHAIN.
- The volume of each tally region must be defined in the `[volume]` section.

Files generated by `[t-dchain]` are listed below.

- The basic input file of DCHAIN: file name is set in the `[t-dchain]` tally.

- Neutron energy spectra with 1968 energy groups below 20 MeV : ***.dtrk
- Nuclear production yields: ***.dyld
- Information on the link to the folder containing the DCHAIN data library is in "dch_link.dat."

Note that when DCHAIN is executed, files of names shown below are deleted.
"yield.out, out-gsdef, out-gamsporg, out-allreg, spd-act.out, angel-data.ang, out-phits, out-dcychains"

7.15 [T-WWG] section

This tally gives parameters used for the [Weight Window] section. The tally serves as a Weight Window Generator (WWG) to automatically obtain effective settings of [Weight Window] for a user-defined virtual space. When using this function, a test calculation using an input file including the [t-wwg] tally must be performed. Long calculations can then be performed using the [Weight Window] section generated from this test calculation. The parameters are determined by the number of particles entering the cells [68]. In the test calculation, PHITS records the number of particles entering the cells specified in [t-wwg], and normalizes their values so as to set the maximum of the weight window to 1. If no particle arrives at a specified cell, the weight window is set to the same value as that in the cell with the minimum fluence. This tally outputs one set of the [Weight Window] section for each type of particle specified by part. If the energy or time mesh is defined in [t-wwg], the [Weight Window] sections are output for each bin. You have to give volumes of each cells in [volume] section.

When you perform the restart calculation or use the sumtally function with an input file including [t-wwg], set ireschk=1 in the [parameters] section.

The [t-wwg] parameters are formatted as follows.

Table 7.44: [t-wwg] parameters (1)

name	value	explanation
mesh =	reg, xyz	Geometry mesh: only reg and xyz can be set; a geometry mesh subsection is required below this option.
part =	all (default), particle name	There is a maximum of six particles in a [t-wwg].
material =	(omitted) all, number of materials	Specify materials for scoring. all : default (same as no definition). To set number of materials, define the material numbers in the next line.
(next line)	2 5 8	Material numbers.
e-type =	1, 2, 3, 4, 5	Energy mesh: an energy mesh subsection is required below this option.
t-type =	1, 2, 3, 4, 5 (omitted)	Time mesh: a time mesh subsection is required below this option.
unit =	1(omitted)	1: [1/cm^2/source]
axis =	eng, reg, t xy, yz, xz wwg	x axis value of output data. 2-dimensional. They are the same as [t-track]. To obtain [Weight Window] section, set axis=wwg. To perform restart calculation, set two (or more) axis parameters, and then set the first axis to reg, eng, or t.
file =	file name	Define file names. This is required by each setting of axis.
resfile =	(omitted, D=file)	Define a file name of the past tally in the restart calculation: even if several axis parameters were defined, specify only one resfile.
factor =	(omitted, D=1.0)	Normalization factor.
title =	(omitted)	Title.
angel =	(omitted)	ANGEL parameters.
rshow =	0 (default), 1, 2, 3	When mesh=xyz and axis=xy,yz,xz, region border (1), material name (2), and region name (3) are plotted using this option. A xyz mesh section must be added below this option.

Only reg and xyz can be set as mesh in this tally, because the parameters in [Weight Window] are defined only for these mesh types. In short, axis should be set wwg to obtain the parameters for [Weight Window]. Although eng, reg, xy, yz, xz, and t can also be set, their results are not related with [Weight Window]. axis= xy, yz, or xz are valid only with rshow=1.

[68] To be precise, it is determined by the fluence of a Monte Carlo particle, i.e., a particle always having weight= 1.

Table 7.45: [t-wwg] parameters (2)

name	value	explanation
ginfo =	0 (default),	No geometry check in the case of gshow or rshow> 0.
	1	Check geometry and draw a two-dimensional view with error information.
	2	Check geometry, draw two-dimensional view, and output a geometry error file (“.err”).
resol =	1 (default)	This option multiplies the region line resolution by a factor of resol with the rshow option set to define the line thickness.
width =	0.5 (default)	The option defines the line thickness.
volume	(omitted)	This option defines the volume for each region for reg mesh. Volume definitions are required below this option. Values defined in [volume] are used in the case of no definition. If special description such as using ( ) is used to specify a region in reg=, its internally defined region number is output in input echo when this volume subsection is not defined.
epsout =	0 (default), 1, 2	When epsout=1, results are plotted into eps files. The eps file is named by replacing the extension with “.eps.” When epsout=2, error bars are also displayed in the eps file, except for the 2-dimensional type (axis=xy, yz, xz, wwg).
ctmin(i) =	(omitted, D=-9999)	Minimum value for *i*-th counter.
ctmax(i) =	(omitted, D= 9999)	Maximum value for *i*-th counter.
trcl =	(omitted)	Coordinate transformation number or definition for r-z or xyz mesh.
gslat =	1(default), 0	1: show lattice boundary in gshow, 0: do not show.
stdcut =	(omitted, D=-1)	Threshold value of STD cut off.

When specifying stdcut, PHITS automatically stop the calculation depending on values of STD (standard deviation). This function is available when stdcut is positive and itall=0,1 is set in [parameters] section. When all relative values of STD of the tally result are smaller than stdcut at the last of one batch, PHITS finishes its calculation. If stdcut in two (or more) tally sections is set, all the results of the tally sections have to satisfy the conditions in order to work the function.

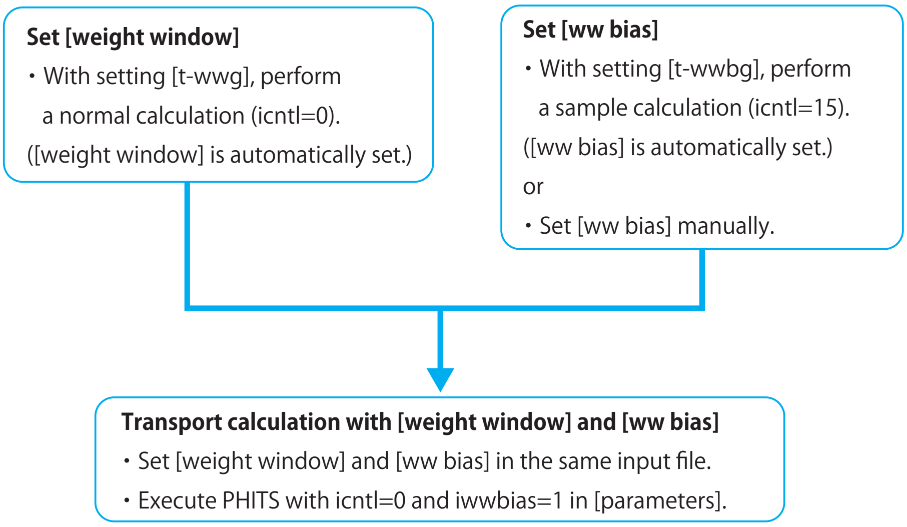
An example output of this tally is as follows.

```
[ Weight Window ]
     mesh =  reg

set:  c71[0.0]  c72[c71+2.13496E-08]  c73[1.25478E-04]
set:  c74[0.0]  c75[c74+9.29569E-09]  c76[2.08987E-04]

    part = neutron
     eng =  2
          1.00000E-03   1.00000E+03

    reg          ww1                    ww2
    1   (8.94092E-05+c71)/c73  (2.08987E-04+c74)/c76
    2   (1.25478E-04+c71)/c73  (1.26817E-04+c74)/c76
    3   (8.53835E-05+c71)/c73  (5.78131E-05+c74)/c76
```

In this sample, the [Weight Window] parameters for neutrons with two energy bins are output. In principle, it is not necessary to change these parameters, but c71 or c74 should be specified when adding a constant value for each Weight Window.

7.16 [T-WWBG] section

This tally gives parameters used for the [ww bias] section. The tally serves as a Weight Window Bias Generator (WWBG) to automatically obtain effective settings of [ww bias], which is used to bias [weight window]. The [ww bias] function is useful when the [weight window] parameters for a certain region are biased after automatically generating the [weight window] section by [t-wwg]. Figure 5.48 shows a flowchart how to perform the transport calculation using [weight window] and [ww bias]. First, generate the [weight window] section by [t-wwg]. If the obtained parameters of [weight window] are enough to effectively use the variance reduction technique, the [ww bias] section is not needed. However, if they are not enough, the variance reduction technique can be used more effectively by biasing the [weight window] parameters for a certain region with [ww bias]. There are two methods to set the [ww bias] section; an automatic method using [t-wwbg], which is a Weight Window Bias Generator (WWBG) and will be explained below, and a manual method by a user. Only one [t-wwbg] section can be set in one input file. [t-wwbg] is effective only when icntl=15 in [parameters]. For details of [ww bias], see Sec. 5.20. When performing a transport calculation with [ww bias] and [weight window], set icntl=0 and iwwbias=1 in [parameters].



Set [weight window]
- With setting [t-wwg], perform a normal calculation (icntl=0).

([weight window] is automatically set.)

Set [ww bias]
- With setting [t-wwbg], perform a sample calculation (icntl=15).

([ww bias] is automatically set.)

or
- Set [ww bias] manually.

Transport calculation with [weight window] and [ww bias]
- Set [weight window] and [ww bias] in the same input file.
- Execute PHITS with icntl=0 and iwwbias=1 in [parameters].

Figure 7.6: The flowchart of the connection calculation between [weight window] and [ww bias].

[t-wwbg] determines the bias values in stages of some cylindrical regions, which are with a central axis on a vector defined by two points (x0,y0,z0) and (x1,y1,z1), shown in Fig. 7.7. Figure 7.7 shows a cross-section view of three different size cylinders. To define the cylindrical regions, the parameters n-mesh, r-mesh, z-mesh, and f-mesh are also required. n-mesh is the number of the cylinders. The differences of radii and heights of the cylinders are given by r-mesh, z-mesh, respectively. In these parameters, the same number of values as the n-mesh must be given. The bias values can be set from the inside of the cylindrical regions in f-mesh. f-mesh must be set of (n-mesh)+1. The last value of f-mesh is the bias value in the outside of the cylinders. The

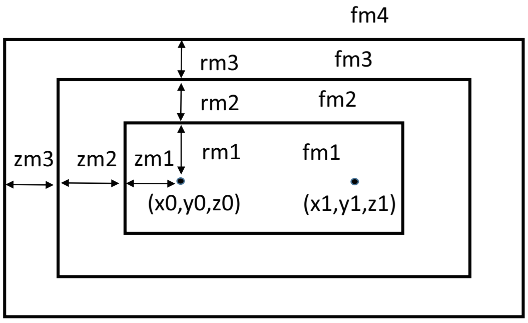


Figure 7.7: Parameters to define cylindrical regions.

cylindrical regions can be defined regardless of the geometry to calculation of the particle transport simulation. Note that if the defined cylindrical regions overlap the outer void given in `[cell]`, the outer void region can be extended by the `r-out` parameter.

The `[t-wwbg]` parameters are formatted as follows.

Table 7.46: `[t-wwbg]` parameters

name	value	explanation
`mesh =`	`reg`	Geometry mesh: only `reg` can be set; a geometry mesh subsection is required below this option.
`part =`	`all` (default), particle name	There is a maximum of six particles in a `[t-wwbg]`.
`e-type =`	`1, 2, 3, 4, 5`	Energy mesh: an energy mesh subsection is required below this option.
`t-type =`	`1, 2, 3, 4, 5` (omitted)	Time mesh: a time mesh subsection is required below this option.
`axis =`	`xy, yz, xz, wwbg`	Restart calculation cannot be performed. When `xy, yz, zx`, the `rshow` option is needed.
`file =`	`file name`	Define file names. This is required by each setting of `axis`.
`angel =`	(omitted)	ANGEL parameters.

`x-txt =`
`y-txt =`
`z-txt =`
	(omitted)
	(omitted)
	(omitted)
		x axis title.
		y axis title.
		z axis title.

`rshow =`	`0` (default), `1, 2, 3`	When `mesh=xyz` and `axis=xy,yz,xz`, region border (1), material name (2), and region name (3) are plotted using this option. A `xyz` mesh section must be added below this option.
`ginfo =`	0 (default), 1 2	No geometry check in the case of `gshow` or `rshow`> 0. Check geometry and draw a two-dimensional view with error information. Check geometry, draw two-dimensional view, and output a geometry error file (".err").

`resol =`
`width =`
	1 (default)
	`0.5` (default)
		This option multiplies the region line resolution by a factor of `resol` with the `rshow` option set to define the line thickness.
		The option defines the line thickness.

`epsout =`	`0` (default), `1`	When `epsout=1`, results are plotted into eps files. The eps file is named by replacing the extension with ".eps."
`x0, y0, z0 =`		x, y, z coordinates of an initial point.
`x1, y1, z1 =`		x, y, z coordinates of a terminal point.
`n-mesh =`	n	The number of cylindrical regions.
`r-mesh =`	rm1,rm2,$\cdots$,rmn	The difference of radii of cylinders. The number of `n-mesh` must be set.
`z-mesh =`	zm1,zm2,$\cdots$,zmn	The difference of heights of cylinders. The number of `n-mesh` must be set.
`f-mesh =`	fm1,fm2,$\cdots$,fmn,fm(n+1)	Bias values in cylindrical regions. The number of `(n-mesh)+1` must be set.
`r-out =`	(omissible, D=0)	Radius to extend the outer void [cm].

The following is an example of `[t-wwbg]`.

Example 58: Example of `[t-wwbg]`.

```
[ T - WWBG ]
     mesh =  reg
      reg = all
     axis = wwbg
     file = wwbg.out
     axis =  yz
     file = wwbyz.out
     part =  neutron
   e-type =    1
       ne =      2
            0.0  1e-3  1.0
       x0 =  0.0
       y0 = -20
       z0 =  50
       x1 =  0.0
       y1 =  20
       z1 =  150
   n-mesh =      3
   r-mesh =   10  10  10
   z-mesh =   10  10  10
   f-mesh =   1.0  0.5  0.1  0.05
    r-out =  1000
```

In this example, the initial and terminal points are (0, -20, 50) and (0, 20, 150), respectively. The three cylindrical regions are defined with the central axis on the vector defined the two points. Figure 7.8 shows a spatial distribution of the bias values given by the example input. The radii of the three cylinders are defined by `r-mesh`, the differences of them are 10cm. The heights of the cylinders increase by 20cm, which is two times of 10cm given by `z-mesh`. Bias values are given as 1.0, 0.5, 0.1, and 0.05 from the inside of the cylindrical regions. The change of the biases is shown in Fig. 7.8.

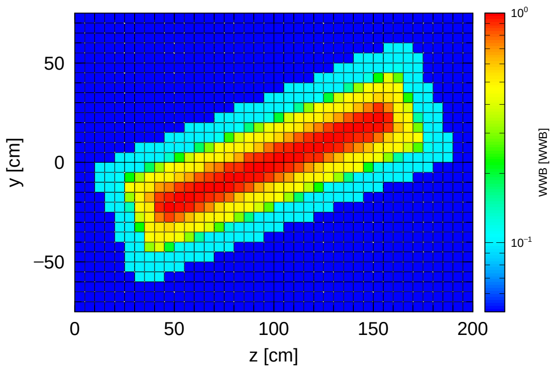


Figure 7.8: Spatial distribution of bias values given by Example 58.

7.17 [T-Volume] section

When you set `mesh=reg` in tally sections, you have to give volumes of each cell to estimate results in the unit of (/volume). PHITS has a function of automatic calculation to obtain the volumes using `[t-volume]`. In this function, PHITS calculates the volumes by performing a Monte Carlo integration with setting the material of each cell to be void and a special source region.

`[t-volume]` is effective only when setting `icntl=14` in `[parameters]` section. Only one `[t-volume]` section can be set in one input file.

The format of parameters in `[t-volume]` is as follows.

Table 7.47: `[t-volume]` parameters

name	value	explanation
`mesh =`	`reg`	Geometry mesh: only `reg` can be set; a geometry mesh subsection is required below this option.
`file =`	`file name`	Define file names.
`resfile =`	(omitted, D=`file`)	Define a file name of the past tally in the restart calculation.
`r-out =`	(omitted, D=`0.0`)	Radius of the outer void. [cm]
`title =`	(omitted)	Title.
`method =`	(omitted, D=0)	Option for Monte Carlo integration. 0: By particle track lengths. 1: By source points.
`s-type =`	1, 2	Source type for volume calculation. 1: Sphere source. The coordinates of the sphere, (`x0, y0, z0`), and its radius, `r0`, have to be defined. 2: Rectangular solid source. The coordinates `x0, x1, y0, y1, z0,` and `z1` have to be defined.
`x0, y0, z0, x1, y1, z1, r0 =`		Coordinates and radius.
`stdcut =`	(omitted, D=-1)	Threshold value of STD cut off.

The source type in the volume calculation is specified by `s-type`. In `s-type=1`, the source generates on a sphere of the center coordinates (`x0,y0,z0`) and the radius `r0` with the inward direction. This is the same condition as `dir=-all` in `s-type=9` of `[source]` section. In `s-type=2`, the source uniformly generates on surfaces of a rectangular solid which are defined by 6 planes, `x=x0, x1, y=y0, y1,` and `z=z0, z1`. Its direction is inward. In either case, you have to set the source region to be large so as to cover the all specified cells.

When specifying `stdcut`, PHITS automatically stop the calculation depending on values of STD (standard deviation). This function is available when `stdcut` is positive and `itall=0,1` is set in `[parameters]` section. When all relative values of STD of the tally result are smaller than `stdcut` at the last of one batch, PHITS finishes its calculation. If `stdcut` in two (or more) tally sections is set, all the results of the tally sections have to satisfy the conditions in order to work the function.

When executing PHITS using `[t-volume]`, the information on the source used in the volume calculation is output in `[source]` of the summary file `file(6) (D=phits.out)`. `[source]` written in the input file of PHITS is not output in the summary file.

An example of [t-volume] is as follows.

Example 59: Example of [t-volume]

```
[ T - V o l u m e ]
  mesh =  reg            # mesh type is region-wise
   reg = 101  102  103  104  105
  file = volume.out      # file name of output for [volume]
s-type =    1            # 1: Sphere source, 2: Rectangular source
    x0 =  0.0            # (D=0.0) x of sphere center
    y0 =  0.0            # (D=0.0) y of sphere center
    z0 =  0.0            # (D=0.0) z of sphere center
    r0 =  50.0           # radius of sphere
```

[t-volume] section should have the setting of mesh=reg. The cell numbers should be specified after reg=. By setting s-type and the related parameters such as x0, you can define the special source region for the volume calculation. Note that the source region should be set to cover all cells specified in reg=, otherwise the calculated volumes would be wrong. On the contrary, if you set an extremely large source region, the statistical uncertainties of the calculated results would be large. The obtained volumes are outputted in a file named by file=. The format of the output file is as follows.

```
[ T - V o l u m e ] off
     mesh =  reg            # mesh type is region-wise
          ....   ....     ....   ....
          ....   ....     ....   ....

[ V o l u m e ]
  non      reg      vol      non
    1      101   5.0370E+02  0.2909
    2      102   2.4727E+03  0.1634
          ....   ....     ....   ....
          ....   ....     ....   ....

# Information for Restart Calculation
          ....   ....     ....   ....
          ....   ....     ....   ....
```

Because the volumes are outputted in the format of [volume] section, the results can be used in the main calculation with icntl=0 by infl command. Here, values of the last column in [volume] are statistical uncertainties (relative values). If the uncertainties are large, you can perform the restart calculation by adding istdev=-1 or -2 in [parameters] section, because the output file of [t-volume] has information for the restart mode.

7.18 [T-Userdefined] section

This tally is used for estimating and outputting physical quantities that cannot be calculated by the other tallies. To use this tally, the user must change “usrtally.f” and re-compile PHITS.

When `[t-userdefined]` is defined in an input file, PHITS calls the subroutine “usrtally” at each moment in the PHITS simulation, i.e., with the same timing used for calling the `dumpall` option. A subroutine for outputting all information is written in the default of “usrtally.f,” and the user can set the output so that only required information is output by revising this file: please see the read-me file or sample input file in “\phits\utility\usrtally\” for more detail.

In `[t-userdefined]`, several parameters can be used: `file`, for specifying a file name; `nudtvar`, for specifying the number of variables `udtvar`(*i*); `udtvar`(*i*)[69], for specifying a numerical value. These parameters can be used in the subroutine ‘usrtally’ without recompiling PHITS.

Table 7.48: `[t-userdefined]` parameters

name	value	explanation
`file` =	file name	Output file names used in usrtally subroutine. A maximum of 50 files can be used (the device numbers of these files are from 151 to 200).
`nudtvar` = `udtvar`(*i*) =	0(default) Numerical value (D=0)	The number of available `udtvar`(*i*). Parameters used in usrtally subroutine: these can be defined upto *i* =`nudtvar`.

Example 60: An example input for `[t-userdefined]`

```
      file = output1.dat            <-its device number is iudtf(1)=151
      file = output2.dat            <-its device number is iudtf(2)=152
   nudtvar = 2                      <-nudtvar
 udtvar(1) = 20.0                   <-udtvar(1)
 udtvar(2) = -10.0                  <-udtvar(2)
```

The following parameters can be used in the `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

[69] This parameter plays the same role as `udtpara`*i* (*i* = 0 – 9) before ver. 3.01. `udtpara`*i* can be also set after ver. 3.02.

(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) nudtvar:
The number of available `udtvar`(*i*). This is given as `nudtvar` in the input file.

(7) udtvar(*i*):
These are numerical values defined as `udtvar`(*i*) in the input file. `udtvar`(*i*) upto *i*=`nudtvar` can be used. If `udtvar`(*i*) is not defined in the input file, it is set to 0.

(8) 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

(9) 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	:	weight of the particle at (x,y,z)

(10) QS:
This is dE/dx for electrons at (x,y,z).

(11) 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.

(12) 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.)

(13) NAME, NCNT(1), NCNT(2), NCNT(3):

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

(14) WT, U, V, W:

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

(15) 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)

(16) 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)

(17) SPX, SPY, SPZ:

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

(18) NZST:
This is charge state of the particle.

(19) 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
13	:	Photon reactions by EGS5
14	:	Electron reactions by EGS5

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	:	x-component of unit vector of momentum
= 2	:	y-component of unit vector of momentum
= 3	:	z-component of unit vector of momentum
= 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)

7.19 [T-Gshow] section

This tally produces graphical geometry output for regions bounded by xyz mesh. These outputs can be obtained without transport calculation using the option icntl=7 in the [parameters] section.

The option of output=10 (gshow=5 for other tallies) is useful for lattice and tetrahedral geometry with very fine mesh.

Table 7.49: [t-gshow] parameters

name	value	explanation
mesh =	xyz	Geometry mesh: only xyz mesh; a geometry mesh subsection is required below this option.
axis =	xy, yz, xz	2-dimensional.
file =	file name	Define file names. This is required by each setting of axis.
output =	1 2 3 4 5 6 7 8 9 10	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 No output Material color in pixel style
resol = width =	1 (default) 0.5 (default)	This option multiplies the region line resolution by a factor of resol. The option defines the line thickness.
title =	(omitted)	Title.
angel =	(omitted)	ANGEL parameters.
sangel =	(omitted)	Special format for ANGEL parameters.
x-txt = y-txt =	(omitted) (omitted)	*x* axis title. *y* axis title.
epsout =	0 (default), 1	Results are plotted into eps files. This eps file is named by replacing the extension into ".eps ".
bmpout =	0(default), 1	Generate Bitmap figure of 2-dimensional tally output. This file is named by replacing the extension with ".bmp." When mesh=xyz, axis and axis=xy, yz, xz, it is available.
vtkout =	0(default), 1	Output the tally results in the xyz-mesh in the input format of ParaView. This file is named by replacing the extension with ".vtk." When mesh=xyz, axis and axis=xy, yz, xz, it is available.
vtkfmt =	0(default), 1	Format of output file for ParaView. 0: ascii, 1: binary.
trcl =	(omitted)	Coordinate transformation number or definition for r-z or xyz mesh.
gslat =	1(default), 0	1: show lattice boundary in gshow, 0: do not show.
ginfo =	0 (default), 1 2	No geometry check in the case of gshow or rshow> 0. Check geometry and draw a two-dimensional view with error information. Check geometry, draw two-dimensional view, and output a geometry error file (".err").

output=7,8 can be used only when the cells in bottom level are themselves in the lattice; the output then gives lattice numbers in the format (4,1,2). For example, Fig. 5.40 in Sec. 5.6.4 is generated by the input shown below.

Example 61: [t-gshow] example

```
[ T - gshow ]
     mesh = xyz
   x-type = 2
       nx = 100
     xmin = -10.
     xmax =  10
   y-type = 1
       ny = 1
           -5.0 5.0
   z-type = 2
    nz    = 100
    zmin  = -10.
    zmax  =  10.
     axis = xz
   output = 8
     file = cell-example6.dat
   epsout = 1
```

7.20 [T-Rshow] section

This tally produces graphical geometry output for regions bounded by `xyz` mesh with color plotting in proportion to physical quantities of the regions. Usually, this tally uses results obtained by a PHITS calculation with `reg` mesh as input data to display the distribution of the physical quantity. These outputs can be obtained without transport calculation using the option `icntl=9` in the `[parameters]` section.

A scale of the color gradation can be changed by setting ANGEL parameters `zlog` and `zlin` to log and linear scales, respectively. Default setting is `zlin`.

Table 7.50: `[t-rshow]` parameters

name	value	explanation
`mesh =`	`xyz`	Geometry mesh: only `xyz` mesh; a geometry mesh subsection is required below this option.
`axis =`	`xy, yz, xz`	2-dimensional.
`file =`	`file name`	Define file names. This is required by each setting of `axis`.

`output =`	`1`
	`2`
	`3`
	`4`
		Region boundary
		Region boundary + material name
		Region boundary + region name
		Region boundary + LAT number

`resol =`
`width =`
	`1` (default)
	`0.5` (default)
		This option multiplies the region line resolution by a factor of `resol`.
		The option defines the line thickness.

`title =`	(omitted)	Title.
`angel =`	(omitted)	ANGEL parameters.
`sangel =`	(omitted)	Special format for ANGEL parameters.

`x-txt =`
`y-txt =`
`z-txt =`
	(omitted)
	(omitted)
	(omitted)
		x axis title.
		y axis title.
		z axis title.

`reg =`
`value`
`reg val`
		Region definition.
		
		Value definition with the same format as volume definition. For details see Sec. 6.1.2.

`iechrl =`	`72` (default)	Maximum number of columns for volume input echo.
`epsout =`	`0` (default)`, 1`	Results are plotted into eps files. This eps file is named by replacing the extension into “.eps ”.
`bmpout =`	0(default), 1	Generate Bitmap figure of 2-dimensional tally output. This file is named by replacing the extension with “.bmp.” When `mesh=xyz, axis` and `axis=xy, yz, xz`, it is available.
`trcl =`	(omitted)	Coordinate transformation number or definition for `r-z` or `xyz` mesh.
`gslat =`	1(default), 0	1: show lattice boundary in `gshow`, 0: do not show.
`ginfo =`	0 (default), 1 2	No geometry check in the case of `gshow` or `rshow`> 0. Check geometry and draw a two-dimensional view with error information. Check geometry, draw two-dimensional view, and output a geometry error file (“.err”).

For example, Figure 7.9 can be obtained by the `[t-rshow]` tally shown below from the example (6) shown in Sec. 5.6.4.

Example 62: `[t-rshow]` example

```
[ T - rshow ]
     mesh = xyz
   x-type = 2
       nx = 100
     xmin = -10.
     xmax =  10.
   y-type = 1
       ny = 1
           -5.0  5.0
   z-type = 2
       nz = 100
     zmin = -10.
     zmax =  10.
     axis = xz
   output = 1
     file = cell-example6-rshow.dat
   epsout = 1
      reg = (201<101[-1 1 0]<1) (201<101[0 1 0]<1) (201<101[1 1 0]<1)
           (201<101[-1 0 0]<1) (201<101[0 0 0]<1) (201<101[1 0 0]<1)
           (201<101[-1 -1 0]<1) (201<101[0 -1 0]<1) (201<101[1 -1 0]<1)
    value
   non     reg      val     # reg definition
    1  1000001   1.0000E+00 # ( 201 < 101[ -1 1 0 ] < 1 )
    2  1000002   2.0000E+00 # ( 201 < 101[ 0 1 0 ] < 1 )
    3  1000003   3.0000E+00 # ( 201 < 101[ 1 1 0 ] < 1 )
    4  1000004   4.0000E+00 # ( 201 < 101[ -1 0 0 ] < 1 )
    5  1000005   5.0000E+00 # ( 201 < 101[ 0 0 0 ] < 1 )
    6  1000006   6.0000E+00 # ( 201 < 101[ 1 0 0 ] < 1 )
    7  1000007   7.0000E+00 # ( 201 < 101[ -1 -1 0 ] < 1 )
    8  1000008   8.0000E+00 # ( 201 < 101[ 0 -1 0 ] < 1 )
    9  1000009   9.0000E+00 # ( 201 < 101[ 1 -1 0 ] < 1 )
```

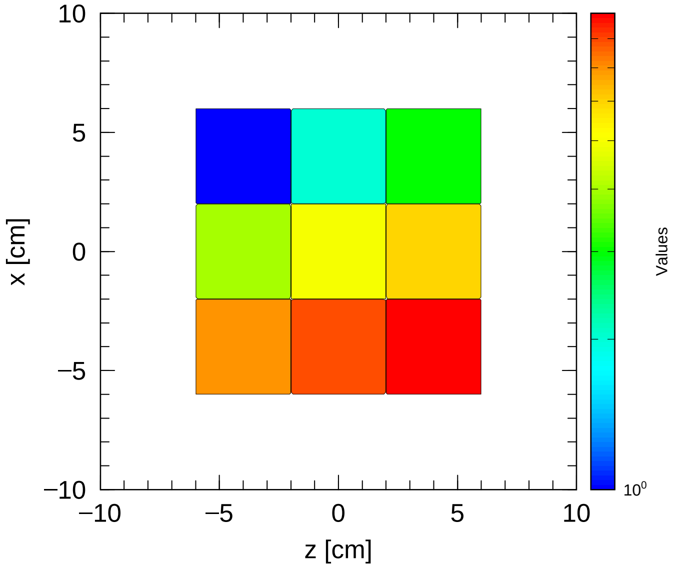


Figure 7.9: Example of `[t-rshow]`.

7.21 [T-3Dshow] section

This tally produces graphical geometry output by 3-dimensional view. This tally can be generated without transport calculation using the option `icntl=11` in the `[parameters]` section.

Table 7.51: `[t-3dshow]` parameters (1)

name	value	explanation

`output =`	`0`
	`1`
	`2`
	`3` (default)
		Draft
		Only region boundary
		Without region boundary
		Region boundary + color

`material =`	(omitted) `all`, number of materials	Specify materials for display. `all` : default (same as no definition). To set number of materials, define the material numbers in the next line. The number of materials can be set as a negative value: in this case the specified materials are not included for display.
(next line)	`2 5 8`	Material numbers.
`x0, y0, z0 =`	(D=0.0)	Coordinates of original point for view point and light source. Center of screen is defined by this point and the view point.

`e-the =`
`e-phi =`
`e-dst =`
	(D=80)
	(D=140)
	(D=w-dst*10)
		View point angle θ [degrees] from z axis.
		Zzimuthal angle for view point ϕ [degrees] from x axis.
		Distance between view point and origin [cm].

`l-the =`
`l-phi =`
`l-dst =`
	(D=e-the)
	(D=e-phi)
	(D=e-dst)
		Light source angle θ [degrees] from z axis.
		Azimuthal angle for light source ϕ [degrees] from x axis.
		Distance between light source and origin [cm].

`w-wdt =`
`w-hgt =`
`w-dst =`
	(D=100)
	(D=100)
	(D=200)
		Width of screen frame [cm].
		Height of screen frame [cm].
		Screen frame distance from origin [cm]. A straight line drawn between the center of the screen frame and the origin crosses the screen surface vertically, and passes through the view point.

`w-mnw =`
`w-mnh =`
`w-ang =`
	(D=100)
	(D=100)
	(D=0.0)
		Number of meshes in horizontal direction.
		Number of meshes in vertical direction.
		Angle of frame [degrees].

`heaven =`	(D=y)	Topside direction; set `x, -x, y, -y, z, -z`.
`mirror =`	(D=0)	=-1; Mirror transformation in horizontal direction.

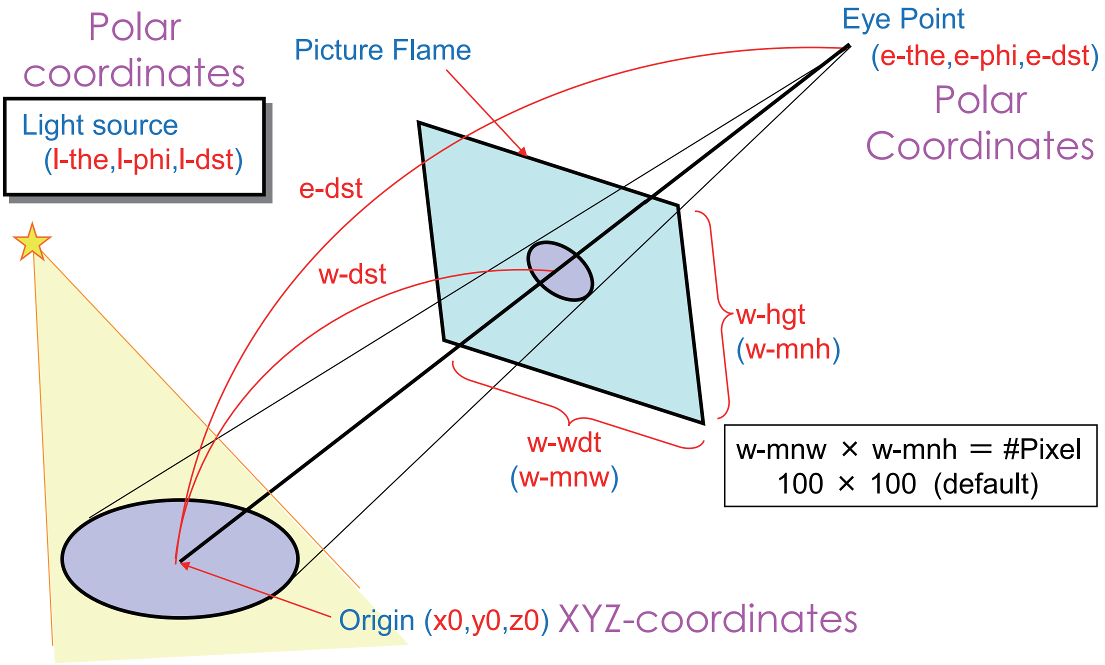


Figure 7.10: 3dshow tally: origin (`x0,y0,z0`), eye point (`e-the,e-phi,e-dst`), light source (`l-the,l-phi,l-dst`), and picture flame (`w-wdt,w-hgt,w-dst`).

Table 7.52: [t-3dshow] parameters (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 = dark =	(D=0.8) (D=0.2)	Brightness limit (1:max, 0:no brightness). Darkness limit (1:no darkness, 0:max).
box = box	(D=0) 10 numbers	Number of penetrations per box; maximum=5. Box definition (see below).
matinbox =	(omitted) all, number of materials	Materials in the box for display. all: default (same as no definition); If number of materials is set, material numbers should be defined in the next line (number of materials cannot be set as negative).
(next line)	2 5 8	Material numbers.
reginbox =	(omitted) all, region numbers	Regions in the box for display. all: default (same as no definition); If the matinbox is defined for a region, the region is not displayed.
resol = width =	1 (default) 0.5 (default)	This option multiplies the region line resolution by a factor of resol. The option defines the line thickness.
file =	file name	Define file names.
title =	(omitted)	Title.
angel =	(omitted)	ANGEL parameters.
sangel =	(omitted)	Special format for ANGEL parameters.
x-txt = y-txt = z-txt =	(omitted) (omitted) (omitted)	*x* axis title. *y* axis title. *z* axis title.
epsout =	0 (default), 1	Results are plotted into eps files. This eps file is named by replacing the extension into ".eps ".
axishow =	(D=1) 0, 1, 2	0: No. Axis not shown. 1: Small axis is shown in the lower-left of the figure. 2: Large axis is shown in the center of the figure.

The definitions of the rules for reg= and reginbox= are the same as that for the region mesh in Sec. 6.1.1.

To saving calculation time, an outer void defined by the radius r-out has been introduced. A larger r-out value must be used when using a large geometry or when placing the light source and view point at long distances. As this new outer void definition can be seen in input echo, input echo cannot be used by icntl=11 as an input for the next calculation.

No shadow is created if the view point and light source are set in the same position.

7.21.1 Box definition

A maximum of five penetration boxes can be defined; defined boxes become transparent. To define a box, three points must first be set as **b**$_0$(x$_0$,y$_0$,z$_0$), **b**$_1$(x$_1$,y$_1$,z$_1$), and **b**$_2$(x$_2$,y$_2$,z$_2$), respectively. The fourth point **b**$_3$ from **b**$_0$ is defined to lie *L* cm in the vertical direction of the plane defined by these three points, i.e., in the (**b**$_2$ - **b**$_0$) direction. In the box definition, a coordinate transformation such as trcl= transform number or trcl = (........) can be used prior to the definition of the points.

This function may fail when a void region is included in the penetration box; in this case, the user should fill the void region with a material of very low density, e.g., air.

The box definition is shown below. Each relation is also shown in Fig. 7.11.

```
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
```

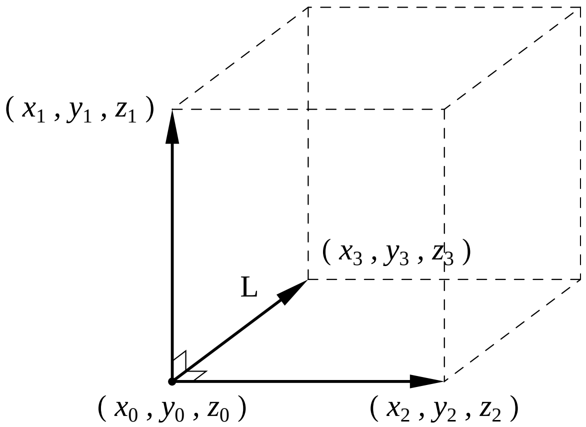

(x_1 , y_1 , z_1)
(x_3 , y_3 , z_3)
L
(x_0 , y_0 , z_0)
(x_2 , y_2 , z_2)

Figure 7.11: Example of box definition.

7.21.2 3dshow example

Example 63: [t-3dshow] example (1)

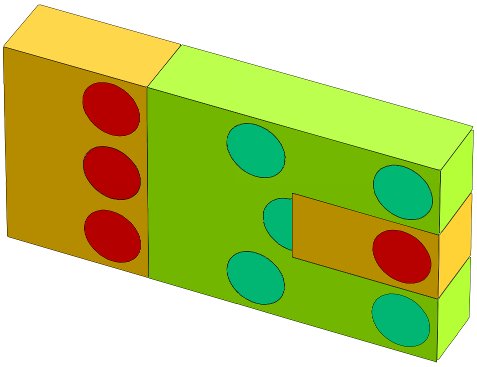
```
[cell]
    1  0 -1 fill=1
    2  0 -41 42 -43 44 -45 46  u=1 fill=5
   22  0 -41 42 -43 44 -45 46  u=1 trcl=(0 0 20) fill=6
   23  like 22 but trcl=(0 0 40) fill = 7
    5  0 -21 22 -23 24 -25 26  u=5 lat=1 fill=3
    6  0 -21 22 -23 24 -25 26  u=6 lat=1 fill= -1:1 0:0 0:0  2 2(0 0 5) 2
    7  0 -21 22 -23 24 -25 26  u=7  fill= -1:1 0:0 0:0  2 3 2 lat=1
    3  1  3.97300E-02  3 u=2
    4  6  4.18280E-02 -3 u=2
   13  5  8.47130E-04 -3 u=3
   14  3  1.23620E-01  3 u=3
   8  -1 +1
[surface]
    1   rpp -15 15 -5 5  -5 55
    21   px    5
    22   px   -5
    23   py    5
    24   py   -5
    25   pz   15
    26   pz   -5
    41   px   15
    42   px  -15
    43   py    5
    44   py   -5
    45   pz   15
    46   pz   -5
    5   rpp -20 20 -5 5 -5 35
    6   rpp -20 20 -5 5 -5 15
    7   rpp -20 20 -5 5 35 55
    3   c/y   0 10  4
```

In the above geometry, the overall body is a rectangular solid with rectangular solid lattices including cylinders on the interior. A graphical plot for the geometry can be created using 3dshow as follows.

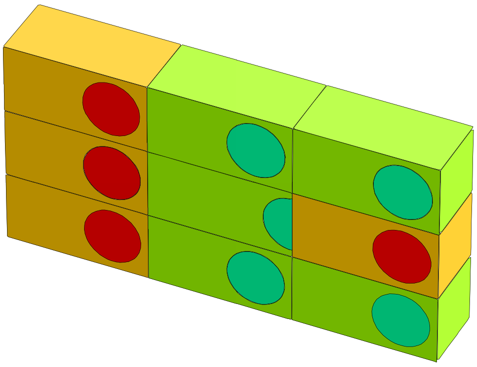
Example 64: [t-3dshow] example (2)

```
[t-3dshow]
   output = 3
   heaven = x
   resol = 2
   width = 0.1
   x0 =  0
   y0 =  0
   z0 = 25
   e-the = 70
   e-phi = 50
   e-dst = 1000
   l-the = 50
   l-phi = 25
   l-dst = 2000
   w-wdt = 60
   w-hgt = 40
   w-dst = 150
   file = dshow.dat
```

The output result is then



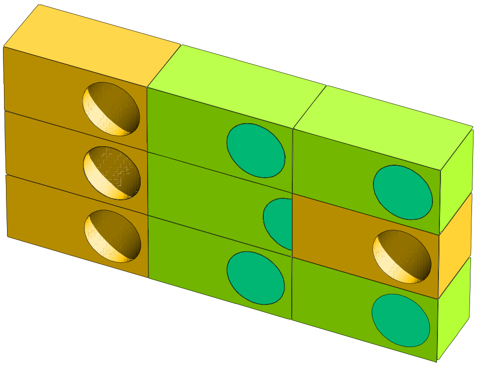
A region boundary can be added using option `line=1` to produce the following `box`, which shows how the lattices are set up:



Setting material number 5 to be transparent and adding shadows using

```
material = -1
            5
  shadow =  2
```

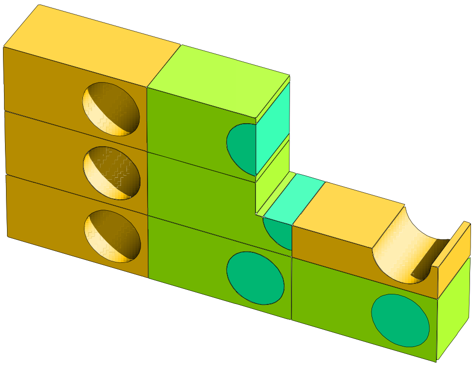
produces the following box:



Next, defining the following box:

```
box = 1
box     0   10   30
      100   10   30
        0   10   100 100
```

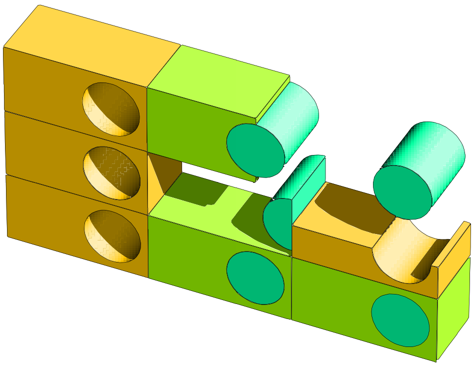
produces a transparent box that shows the inside of its body:



As a final example, by adding

```
reg = ( 3 < 6[0 0 0] )
matinbox = 1
           6
```

the regions defined by `reg = (3 < 6[0 0 0] )` become transparent and material number 6 becomes visible:



Any number of complex structure types can be created by combining the above options.

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 folder “\src\” and the execute file “dump_a.exe” in Windows system is include in the folder “\bin\.”

File 3: dump-a.f

```
************************************************************************
*                                                                      *
*     This program exchanges the binary data and the ascii data        *
*     of dump file.                                                    *
*                                                                      *
*       modified by K.Niita on 2005/08/15                              *
*                                                                      *
*                                                                      *
*                                                                      *
*                                                                      *
************************************************************************
      implicit real*8 (a-h,o-z)
*-----------------------------------------------------------------------
      dimension isdmp(0:30)
      dimension jsdmp(0:30)
      data isdmp / 31*0 /
      data jsdmp / 31*0 /
      character chin*80
      character chot*80
      logical exex
      character dmpc(30)*4
      data dmpc / '  kf','   x','   y','   z','   u','   v','   w',
     &           '   e','  wt','  tm','  c1','  c2','  c3',
     &           '  sx','  sy','  sz','  n0','  nc','  nb','  no',
     &           '    ','    ','    ','    ','    ','    ',
     &           '    ','    ','    ','    '/
      dimension dmpd(30)
      dimension dmpp(30)
      data dmpp /  2112.,   0.0,   0.0,   0.0,   0.0,   0.0,   1.0,
     &              100.,   1.0,   0.0,   0.0,   0.0,   0.0,
     &              0.0,    0.0,   0.0,   0.0,   1.0,   1.0,   1.0,
     &              0.0,    0.0,   0.0,   0.0,   0.0,   0.0,   0.0,
     &              0.0,    0.0,   0.0/
*-----------------------------------------------------------------------
         in = 5
         io = 6
         id = 20
         ia = 21
         iserr = 0
*-----------------------------------------------------------------------
*        user program frag : 0 => no,  1 => with user program
*-----------------------------------------------------------------------
         iuser = 0
*-----------------------------------------------------------------------
*        read ascii or binary frag
*-----------------------------------------------------------------------
         write(io,*) ' ** 0 => read binary to ascii'
         write(io,*) ' ** 1 => read ascii  to binary'
         read(in,*,end=993) iasb
*-----------------------------------------------------------------------
*        read the name of input dump file
*-----------------------------------------------------------------------
         write(io,*)
         write(io,*) ' ** put the file name of input dump file'
         read(in,'(a80)',end=998) chin
            inquire( file = chin, exist = exex )
            if( exex .eqv. .false. ) then
              write(io,*) ' ** Error : the file does not exist'
              goto 999
            end if
            if( iasb .eq. 0 ) then
              open(id, file = chin,
     &             form='unformatted',status = 'old' )
            else
              open(id, file = chin,
     &             form='formatted',status = 'old' )
            end if
*-----------------------------------------------------------------------
*        read the number of data and data sequence
*-----------------------------------------------------------------------
         write(io,*)
         write(io,*) ' ** put the number of data in a record'
         read(in,*,end=997) isdmp(0)
         write(io,*)
         write(io,*) ' ** put the ID numbers of data in a record'
         read(in,*,end=996) ( isdmp(i), i = 1, isdmp(0) )
            do k = 1, isdmp(0)
               if( isdmp(k) .gt. 20 .or.
     &             isdmp(k) .le.  0 ) goto 992
                  jsdmp(isdmp(k)) = k
            end do
            write(io,*)
            write(io,'('' # dump data : '',30(a4))')
     &         ( dmpc(isdmp(j)), j = 1, isdmp(0) )
*-----------------------------------------------------------------------
*        read the name of output dump file
*-----------------------------------------------------------------------
         write(io,*)
         write(io,*) ' ** put the file name of output'
         read(in,'(a80)',end=998) chot
            inquire( file = chot, exist = exex )
            if( exex .eqv. .true. ) then
              write(io,*)
              write(io,*) ' ** Warning : the file already exists'
              write(io,*) ' ** Do you want to overwrite ?'
              write(io,*) ' ** Yes <= 0, No <= 1'
              read(in,*,end=995) iyes
              if( iyes .ne. 0 ) goto 999
            end if
            if( iasb .eq. 0 .or. iuser .ne. 0 ) then
              open(ia, file = chot,
     &             form='formatted',status = 'unknown' )
            else
              open(ia, file = chot,
     &             form='unformatted',status = 'unknown' )
            end if
*-----------------------------------------------------------------------
*        read the number of records to read
*-----------------------------------------------------------------------
         write(io,*)
         write(io,*) ' ** put the number of records to read'
         write(io,*) ' ** all <= 0, or positive integer'
         read(in,*,end=994) irec
*-----------------------------------------------------------------------
*        start reading the data
*-----------------------------------------------------------------------
         write(io,*)
         write(io,*) ' ** start read and write the data'
*-----------------------------------------------------------------------
         jrec = 0
  100    jrec = jrec + 1
         if( irec .gt. 0 .and. jrec .gt. irec ) goto 500
  687       continue
            if( iasb .eq. 0 ) then
               read(id,end=688,err=690)
     &         ( dmpd(isdmp(k)), k = 1, isdmp(0) )
            else
               read(id,'(30(1p1d24.15))',end=688,err=690)
     &         ( dmpd(isdmp(k)), k = 1, isdmp(0) )
            end if
            goto 689
  688       if( irec .gt. 0 ) then
               rewind id
               goto 687
            else
               goto 500
            end if
  690       continue
            iserr = iserr + 1
            write(io,'(''  ** Error in dump file no ='',i5)') iserr
            goto 687
  689       continue
*-----------------------------------------------------------------------
*     user program here
*-----------------------------------------------------------------------
      if( iuser .ne. 0 ) then
            do k = 1, 20
               if( jsdmp(k)  .gt. 0 ) dmpp(k) = dmpd(k)
            end do
               kf = nint( dmpp(1) )
               x  = dmpp(2)
               y  = dmpp(3)
               z  = dmpp(4)
               u  = dmpp(5)
               v  = dmpp(6)
               w  = dmpp(7)
               e  = dmpp(8)
               wt = dmpp(9)
               t  = dmpp(10)
               n1 = nint( dmpp(11) )
               n2 = nint( dmpp(12) )
               n3 = nint( dmpp(13) )
               sx = dmpp(14)
               sy = dmpp(15)
               sz = dmpp(16)
               n0 = nint( dmpp(17) )
               nc = nint( dmpp(18) )
               nb = nint( dmpp(19) )
               no = nint( dmpp(20) )
      end if
*-----------------------------------------------------------------------
*     write data on the file
*-----------------------------------------------------------------------
      if( iuser .eq. 0 ) then
            if( iasb .eq. 0 ) then
               write(ia,'(30(1p1d24.15))')
     &         ( dmpd(isdmp(k)), k = 1, isdmp(0) )
            else
               write(ia)
     &         ( dmpd(isdmp(k)), k = 1, isdmp(0) )
            end if
      end if
*-----------------------------------------------------------------------
         goto 100
*-----------------------------------------------------------------------
*        end of process
*-----------------------------------------------------------------------
  500    continue
         write(io,*)  ' ** end of read and write the data'
         write(io,'(''  ** number of processed records is '',
     &         i8)') jrec-1
         write(io,*)
         close( id )
         close( ia )
         goto 999
*-----------------------------------------------------------------------
```

```
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 6.4 6.5.

```
 ** 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 File 3. 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 4.4), 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/u 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 Region error check

In developing a complex geometry, it is diffcult to define the geometry without mistakes such as double-defined or undefined regions. Correspondingly, a function for automatically detecting double-defined or undefined regions was implemented after version 2.67. This geometry check function works when specifying a tally for generating the two-dimensional view of a geometry: such tallies include `[t-gshow]`, `[t-rshow]`, and other tallies involving setting `axis=xy,yz,xz` or the `gshow` (`icntl=8`) or `rshow` (`icntl=10`) options.

The `ginfo` parameter is assigned to each tally to control this geometry check function. `ginfo` is set =`0` for no geometry check, =1 to check the geometry and draw a two-dimensional view with error information (see Fig. 10.1), and =`2` to check geometry, draw a two-dimensional view, and output a geometry error file that specifies the xyz coordinates of the error location. The default value of ginfo for tallies depicting a two-dimensional geometry is 2.

Figure 10.1 shows an example of a two-dimensional view with geometry errors. Double-defined regions are painted in black, while undefined regions are in purple. When an undefined region is detected, its surrounding regions may disappear from the figure.

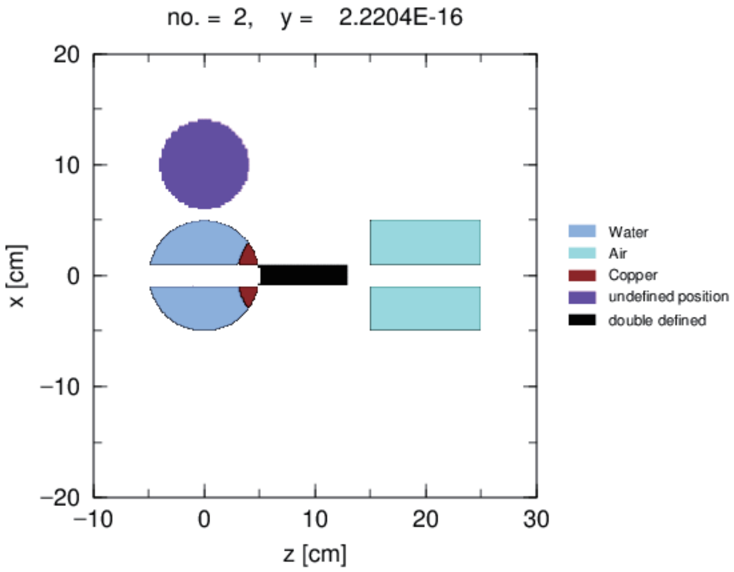


Figure 10.1: Output including geometry errors.

When `ginfo=2` and PHITS detects a geometry error, a geometry error file named "***.err" is output (*** indicates the original tally output file name without the extension). In this file, the (x, y, z) coordinates together with the overlapping cell numbers are written as follows:

```
Errors of cell definition in EPS Page No. =   1
Overlapped Cell IDs  x, y, z  coodinates
(Cells 0       0  indicate undefined region)
     100     102    -4.847761E+00  1.234568E-11 -1.211940E+00
       0       0    -4.241791E+00 -2.500000E+00 -8.079602E-01
```

The first line indicates that cell numbers 100 and 102 overlap at the point `x=-4.847761E+00, y=1.234568E-11, z=-1.211940E+00`. The second line indicates that an undefined region is detected at the point `x=-4.241791E+00, y=-2.500000E+00, z=-8.079602E-01`.

Geometry errors can easily be found using this geometry check function. Note that the function can detect geometry error only when an error occurs on the grid points of the `xyz` mesh of the tally, while geometry errors outside the tally region cannot be detected. Even in the tally region, a small error region might also remain undetected if the region does not contain a grid point.

11 Compilation of PHITS

PHITS must be re-compiled by yourself for some circumstances; e.g. you need to extend the maximum memory allowed to be used by PHITS, or you would like to use your own source generation or tally programs. Our recommended Fortran compilers are Intel Fortran 11.1 (or later) and gfortran (4.8, 7.0 or later). [70] Table 11.1 summarizes the combination of the Fortran compiler and OS by which we have succeeded in compiling PHITS for four different parallel options: single mode, MPI, OpenMP, and hybrid of MPI and OpenMP.

Table 11.1: The available fortran compilers for each OS

OS	Intel Fortran				gfortran			
	single	MPI	Open MP	hybrid	single	MPI	Open MP	hybrid
Windows	✓	✓	✓	✓	✓	✓		
Mac	✓		✓		✓			
Linux	✓	✓	✓	✓	✓	✓		

11.1 Compilation on Windows OS

11.1.1 Compilation using makefile

The PHITS code can be compiled using the “make” command. For this purpose, the “makefile” file in “\src” folder should be revised to fit for the host computer. For example, to compile using the gfortran compiler on Windows OS, the `ENVFLAGS` written in “makefile” should be set to `WinGfort`. To execute PHITS via parallel computing using MPI or OpenMP, it is necessary to set `USEMPI` or `USEOMP`, respectively, to `true`. The name of the executable file is given as “phits_XXX.exe,” where XXX is a parameter set to `ENVFLAGS`. For example, when `ENVFLAGS=WinGfort`, the name is phits_WinGfort.exe.

The compiler options given in “makefile” are simply examples, and users may need to change the setting to fit their own computer environment.

The attached “makefile” can be used for GNU make. If an error is encountered with the “make” command, try the “gmake” command instead.

To use gfortran for Windows, the latest installer version can be downloaded from “Downlaod Latest Version” on the following web site:

- MinGW-w64 - for 32 and 64 bit Windows
(`https://sourceforge.net/projects/mingw-w64/files/?source=navbar`)

Running the downloaded installer, select “x86_64” or “i686” for 64-bit or 32 bit PC, respectively, as “Architecture.” It is unnecessary to change other parameters during the installation process.

The followings are the procedure of how to compile PHITS with the gfortran compiler.

(1) Revise “makefile” in “\phits\src\” folder: set `ENVFLAGS=WinGfort`. Note that OpenMP cannot be activated.

(2) Open a command prompt by clicking “mingw-w64.bat” in the gfortran installation folder.

(3) Use the “cd” command and go to “\phits\src\” folder. Type “mingw32-make” to compile PHITS.

The compiled PHITS by gfortran can be used as follows:

(1) Copy “phits_WinGfort.exe” to “\phits\bin\.”

(2) Open “phits.bat” in “\phits\bin\”, and add

```
set PATH=C:\Program Files\mingw-64\...
```

[70] Note that PHITS compiled by Intel fortran is much faster than that compiled by gfortran, so we recommend to use Intel Fortran if it is available. Intel Fortran is free for students. Please check the link in more detail: `https://software.intel.com/en-us/qualify-for-free-software/student`

written in the 2nd line of "mingw-w64.bat," . Then set "PHITS_EXE" to the PHITS executable file compiled by gfortran, i.e., "phits_WinGfort.exe."

(3) Run PHITS via Windows explorer using "sendto → PHITS" command.

(4) After finishing the PHITS calculation, some warnings, such as "Note: The following floating-point exceptions are signaling: IEEE_DENORMAL." , may show up, but can be be ignored.

11.1.2 Compilation using Microsoft Visual Studio with Intel Fortran

The "\phits\bin\" folder contains a project file ("phits-intel.proj") required for compiling PHITS using Microsoft Visual Studio coupled with Intel Fortran. PHITS can be compiled using these files as follows:

(1) Double-click the "phits-intel.proj" file [71].

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

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

(4) Execute the project in release mode.

(5) Type `file=input_file_name` in the console window.

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

To compile PHITS in the shared-memory parallel computing mode, it is necessary to add "/Qopenmp" in the additional option window (see project → property → Fortran → command line) before building "phits-intel.proj."

To use the compiled PHITS using the "sendto" command, rewrite the environmental variable "PHITS_EXE" written in "phits.bat,", e.g.,

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

Although it is possible to use the "sendto" command to rename the compiled PHITS to the original file name, e.g., "phits282_win.exe," the original file should not be deleted, as it is required for PHITS version update.

11.2 Compilation on Mac OS

The PHITS code can be compiled using the "make" command. For this purpose, the "makefile" file in "\src" folder should be revised to fit for the host computer. For example, to compile using Intel Fortran Compiler on Mac, the `ENVFLAGS` written in "makefile" should be set to `MacIfort`. To execute PHITS via parallel computing using MPI or OpenMP, it is necessary to set `USEMPI` or `USEOMP`, respectively, to `true`. The name of the executable file is given as "phits_XXX.exe," where XXX is a parameter set to `ENVFLAGS`. For example, when `ENVFLAGS=MacIfort`, the name is phits_MacIfort.exe. If `USEOMP` is also set to `true`, the name would become phits_MacIfort_OMP.exe. The compiler options given in "makefile" are simply examples, and users may need to change the setting to fit their own computer environment.

The attached "makefile" can be used for GNU make. If an error is encountered with the "make" command, try the "gmake" command instead.

11.3 Compilation on Linux OS

The PHITS code can be compiled using the "make" command. For this purpose, the "makefile" file in "\src" folder should be revised to fit for the host computer. For example, to compile using Intel Fortran Compiler on Linux, the `ENVFLAGS` written in "makefile" should be set to `LinIfort`. To execute PHITS via parallel computing using MPI or OpenMP, it is necessary to set `USEMPI` or `USEOMP`, respectively, to `true`. The name of the executable file is given as "phits_XXX.exe," where XXX is a parameter set to `ENVFLAGS`. For example, when `ENVFLAGS=LinIfort`, the name is phits_LinIfort.exe. If `USEOMP` is also set to `true`, the name would become

[71] This file may be automatically updated when a new version of Visual Studio or Intel Fortran is used, and it cannot be opened by an older version of Visual Studio (before 2005) and/or Intel Fortran (before 11.1).

phits_LinIfort_OMP.exe. The compiler options given in “makefile” are simply examples, and users may need to change the setting to fit their own computer environment.

The attached “makefile” can be used for GNU make. If an error is encountered with the “make” command, try the “gmake” command instead.

12 Additional explanation for parallel computing

There are two types of parallel computing: distributed-memory parallel computing using the MPI protocol, and the shared-memory parallel computing using OpenMP architecture. Parallel PHITS calculation can be performed using either of these methods, or by using a hybrid method. To execute distributed-memory parallel computing, the MPI protocol must be installed in the user's computer; by contrast, there is no additional software required for shared-memory parallel computing. Note that when the same number of CPU core is set, the computational time for the distributed-memory parallel computing is often shorter than for the shared-memory parallel computing. Each executable file must be generated according to the type of the parallel computing employed: see Sec. 11 for details. Fortran compilers recommended to make the executable file are Intel Fortran Compiler 11.1 (or later) and gfortran 4.7 (or 4.8) for the distributed-memory parallel computing, while only the Intel compiler can be used for the shared-memory parallel computing.

In distributed-memory parallel computing, jobs are distributed to each CPU core in batch units. When all jobs assigned to each core are finished, the main core gathers their results. In this mode, all cores individually use an amount of memory equivalent to that used in single processing; thus, the total RAM memory in the computer system must be larger than the product of the memory used for single processing and the core number. As a result, this type of parallel computing is not suitable for calculations requiring a large amount of memory, for instance, calculations involving voxel phantoms.

In shared-memory parallel computing, jobs are distributed to each CPU core in the units of history. In this case, the cores share a large portion of the memories used by PHITS aside from those defined as "thread private" variables. Thus, the memory required in this parallel computing mode is nearly the same as that required for single processing. One disadvantage of shared parallel computing relative to memory-distributed computing is its slower computational time as a result of the competition for access to shared memories. This disadvantage becomes very important for calculations in which memories are frequently updated, such as those using the `[t-sed]` tally.

12.1 Distributed memory parallel computing

12.1.1 Installation of MPI protocol

It is necessary to install a MPI protocol for executing the distributed-memory parallel version of PHITS. In this manual, the installation procedure of MPICH2 into a Windows PC is described, which is necessary for using the MPI version of PHITS included in the PHITS package, i.e. "c:\phits\bin\phitsXXX_win_mpi.exe." For other cases, please set up your machine environment with MPI protocol by yourself.

MPICH2 is mainly developed in Argonne National Laboratory. For easy installation, we recommend to download its version 1.4 from `http://www.mpich.org/static/downloads/1.4/` because the Windows installer is provided in the version. For 64-bit and 32-bit Windows, please select "mpich2-1.4-win-x86-64.msi" and "mpich2-1.4-win-ia32.msi," respectively. During the installation, you have to select "C:\Program Files\MPICH2" (default) as the installation folder, and select "Everyone" as users. You can input any "Passphrase," but you have to set the same "Passphrase" for each Windows PC if you use multiple machines for the MPI calculation. After the installation of MPICH2, you have to right-click "smpd-install.bat" contained in "\phits\document\mpi\", and execute it as the administrator.

12.1.2 Execution of MPI version of PHITS on Windows

For executing the MPI version of PHITS on Windows, you have to write `$MPI=`M before the 1st section of your input file, where M is the number of parallelization. For example, you have to write

```
$MPI = 4
```

if you want to run PHITS as fast as possible in your computer with 4 CPU cores. Note that the actual number of processing elements (PEs) is $M + 1$, where the additional element is used for controlling each process.

Then, you have to right-click the input file, and send to "phits" in the same manner as executing the single version of PHITS. You have to input your username and password when you first-time run the MPI version of PHITS. If you write both `$OMP` and `$MPI` in your input file, the command written behind is superseded.

12.1.3 Execution of MPI version of PHITS using command line

For Linux, Mac, or multiple Windows PCs connected via network, you have to use command line for executing the MPI version of PHITS. For example, you have to type

```
mpirun -np 5 phits_LinIfort_MPI.exe
```

where "mpirun" is the executable file name of your MPI protocol, "phits_LinIfort_MPI.exe" indicates the PHITS-executable file name and the number of processing elements (PEs) is set following "-np." This command can be sent using a parallel computing submission protocol such as "qsub," in which case the name of the PHITS input file should be written in a text file named `phits.in` with the first line given as

```
file = input_file_name
```

where "input_file_name" is the name of the PHITS input file. This rule is only effective for distributed memory parallel computing. `file=phits.in` can also be written in the first line, with the contents of the PHITS input file added following the second line of `phits.in`: please see Sec. 2.3.2 for further details.

12.1.4 Adjustment of maxcas and maxbch

In distributed-memory parallel computing, jobs are distributed to each CPU core in units of batch. Hence, the number of batches (`maxbch`) should be a multiple of PE−1 (one of the PEs is used for control). If not, PHITS will automatically change `maxbch` to a multiple of PE−1 and adjust the number of histories per batch, `maxcas`, to make the total history number equivalent to the value set in the input file. In this case, relevant comments will be output at the end of the input echo.

In restart mode (`istdev<0`), adjustment of `maxcas` is not performed, as this value should be set to the same value as was written in past tally results.

12.1.5 Treatment of abnormal ending

When PHITS stops as the result of an abnormal end in a PE, the PE is removed from the operation and a total result by remaining PE is given as the final result. In this case, the user should check the `ncut` file, which will be incomplete.

12.1.6 ncut, gcut, pcut, and dumpall file definition in PHITS

For the parallel calculation, `ncut, gcut`, and `pcut` can be defined normally in an input file as

```
file(12) = temp/ncut.dat
```

In 1 PE calculation, the specified `ncut.dat` is written normally; in multi PE calculation, however, `ncut.dat` is written separately in each node as

```
/wk/j9999/temp/ncut.dat
```

where "j9999" is the user-name, which is read-in automatically from the environmental variable "LOGNAME." By default, the user-name is placed in the "LOGNAME" in a UNIX system.

Before parallel calculation, a "j9999" directory should be created under the "/wk" directory for each node. To create an `ncut` file in a directory not named using the user-name, the environmental variable "LOGNAME" should be changed before parallel calculation. In the case, the user should confirm the existence of the directory that they specified under "/wk."

`inpara, igpara`, and `ippara` are used as writing options. By default, they each have values of zero: re-setting the value= 1 gives the output files IP numbers as follows:

`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 number.

Setting inpara, igpara, or ippara=3 averts addition of the default file path

```
/wk/j9999/
```

In each case, setting the value= 3 places the IP number at the end of the file, as in the case = 1.

12.1.7 Read-in file definition in PHITS

The read-in files used in PHITS are Decay-Turtle source files. Although these files generally have only a small effect on network traffic, they can occasionally reach sizes as large as 100 MB and, if there is a read-in for every event, their effect to network traffic can increase. Thus, the Decay-Turtle data file should be copied and placed in each PE as "\wk\j9999\turtle\sours.dat" and then defined as file=/wk/j9999/turtle/sours.dat in the PHITS input.

12.2 Shared-memory parallel computing

12.2.1 Execution

Except when Linux or PHITS is directly specified by a command input, PHITS calculation using shared-memory parallel computing can be executed by adding "$OMP=*N*" (*N* is the number of CPU cores to be used) before the first section in a PHITS input file. When $N = 0$, all cores in the computer are used. If $N = 1$ is set, the parallel computing is not used. From version 2.73, the installed executable file of the OpenMP version is available only on 64-bit Windows systems. Note that the function of $OMP is available only for using phits.bat(on Windows OS) or phits.sh(on Mac OS).

In the case when Linux or PHITS is directly specified by a command input, PHITS with shared-memory parallel computing can be executed using the following command:

```
phits_LinIfort_OMP.exe < phits.inp
```

where phits_LinIfort_OMP.exe indicates a PHITS-executable file compiled using the OpenMP option and "phits.inp" is the PHITS input file. Any name can be used for the PHITS input file, as the file name restriction to phits.in is not valid in this case. To specify the number of cores to be used for parallel computing, the environment variable "OMP_NUM_THREADS" should be defined. Note that this variable should be set to equal the real number of CPU-cores, not the number of total threadds, when using hyper-threading technology, as parallel computing using this technology does not work in PHITS calculation. The environment variable can be changed as follows:

```
export OMP_NUM_THREADS=8
```

Note that in hybrid parallel computing the environment variables need to be set individually on each node.

From version 2.73, the installed executable OpenMP version file is available only for 64-bit Windows systems, as an error owing to heap memory shortage sometimes occurred on Windows OS when the executable file of the OpenMP version for 32-bit was used with many cores. However, this error is avoided by using the executable file for 64-bit Windows.

12.2.2 Important notices for shared-memory parallel computing

Using only one core for memory-shared parallel computing in PHITS takes approximately twice as much time as is needed for single processing. Therefore, it is useless to select memory-shared parallel computing mode for a computer with only one or two cores.

Segmentation errors occur in executing memory-shared parallel computing PHITS on Linux may be caused by the overuse of the stack memory; in this case, the stack size must be increased using the following command:

```
export OMP_STACKSIZE=1G
```

which sets the stack size to 1 GB.

In principle, the results obtained from single processing and shared-memory parallel computing should be the same. Thus, the user should inform us (phits-office) if inconsistencies are found between the results obtained by these two modes, as these may represent bugs in the programming.

13 FAQ

13.1 Questions related to parameter setting

Q1.1 Input file that works before ver. 2.88 does not work after that version.

A1.1 Several revisions were made in terms of the input file format after ver. 2.89 to avoid frequently-occurring mistakes. In general, it is not necessary to change input file when PHITS is updated, but in some cases, it is necessary. If you encounter an error after the update, please check the following points.

(1) After ver. 2.89, 'c' cannot be used as a comment remark in [material] section in the default setting. You have to change the comment remarks c in [material] to $ or #, or set icommat=1 in [parameters].

(2) After ver. 2.93, low energy neutrons are transported using nuclear data library in the default setting. Thus, you may encounter an error "There is no cross section table(s) in xsdir" even when you want to transport only photon and electrons. In that case, you have to set nucdata=0 in [parameter] to disable the use of nuclear data library.

(3) After ver. 2.96, it is not allowed to define two or more [parameter] sections in an input file.

(4) After ver. 2.96, unnecessary tallies are automatically disabled, depending on the icntl parameter. Consequently, set: and infl: commands are ignored when they are written in the disabled sections. When set: or infl: commands are ignored, PHITS outputs warning.

Q1.2 Electrons and positrons are not transported.

A1.2 Transport of electrons and positrons are ignored in the default setting because they are time consuming. If you want to transport them, you have to activate EGS5 mode by setting negs=1 in [parameters].

Q1.3 What nuclear reaction model settings gives the most accurate result?

A1.3 In general, the default models give the best results in most cases. However, it is desirable to activate JQMD-2.0 (irqmd=1) for precisely simulating nucleon-nucleon interactions, and SMM (ismm=1) for precisely estimating the residual nuclide yields after high-energy nuclear reactions, though they are time consumptive.

Q1.4 What kind of simulation is the event-generator mode suited to?

A1.4 Event generator mode is best suited to simulations in which it is necessary to obtain event-by-event information, e.g., detector response calculations and the design of semi-conductor devices. It is also useful for simulations in which the energies and types of charged particles produced by low-energy neutron interactions must be determined. Specifically, event generator mode is generally most suited to simulations using [T-Deposit], [T-LET], [T-SED], [T-Yield], and/or [T-Product] tallies. On the other hand, it is not suited to simulations only using [T-Track] and/or [T-Cross] tallies, such as shielding calculations. See "5.2.22 Event Generator Mode" for further details.

Q1.5 When should the mode for statistical uncertainty (the setting of istdev) be changed?

A1.5 We generally recommend using the history variance mode (istdev=-2 or 2), in which the statistical uncertainty depends on the total history number (maxcas*maxbch), except in the case of shared-memory parallel computing, in which only the batch variance mode (istdev=-1 or 1) can be selected. However, the computational time occasionally becomes extremely long in history variance mode, especially in for tallies using a large number of memories, e.g., an xyz mesh tally with very fine structure. When performing a PHITS calculation under such conditions, please switch to batch variance mode and set maxbch to be greater than 10.

Q1.6 Is it possible to use nuclear data libraries other than JENDL-4.0?

A1.6 Yes. PHITS can use nuclear data libraries that are written in the ACE format, i.e., the MCNP format. The following are instructions on how to use a new nuclear data library in PHITS:

(1) using a text editor, open the "xsdir" file contained in the package of the new nuclear data library and copy all nuclear data addresses (e.g., 1001.80c 0.999167 xdata/endf71x/H/1001.710nc 0 1 4 17969 0 0 2.5301E-08);

(2) using a text editor, open the file “xsdir.jnd” included in the PHITS “data” folder and paste the copied addresses at the end of the file;

(3) create a new folder with the appropriate name written in the address file in the PHITS “XS” folder and copy the data files from the new nuclear data library to the created folder;

(4) explicitly specify the library ID in the `[material]` section of your PHITS input file (e.g., 1H.80c or 1001.80c for the above example). If you do not specify the library ID, PHITS automatically finds the data library of the previously written nucleus in “xsdir.jnd” and, therefore, JENDL-4.0 is selected when the data for the nucleus is available.

Q1.7 How can neutron fluxes emitted from photo-nuclear reactions with low statistical uncertainty be found?

A1.7 The photonuclear cross section can be biased using the `pnimul` parameter. For example, the probability of photo-nuclear reaction is doubled when `pnimul` is set `=2.0` and the weights of secondary particles emitted from the photo-nuclear reaction are 0.5. If a very high value of `pnimul`, e.g., above 100.0, is selected, the photon fluxes might be altered; therefore, it should be confirmed that the photon fluxes are not significantly altered as a result of changing this parameter.

Q1.8 I cannot restart PHITS simulation as a result of an error related to the inconsistency of a tally. Why is this?

A1.8 This occurs as the result of the loss of significant digits in the PHITS input file. In this case, set `ireschk=1` to induce PHITS to skip the consistency check.

13.2 Questions related to errors occurred in compiling or executing PHITS

Q2.1 I got an error in compiling PHITS. How can I correct this to finish compiling?

A2.1 The PHITS office recommends using Intel Fortran Compiler 11.1 (or later) and gfortran 4.7 (or later). However, Windows gfortran 4.9 - 5.4 cannot compile PHITS properly. The recommended compiler is Intel Fortran because PHITS compiled by Intel Fortran is much faster than that by gfortran. See section 11 for further details.

Q2.2 A segmentation fault occurred during the execution of PHITS.

A2.2 This might have occurred as a result of an overflow of the memory used in PHITS. In this case, it may be necessary to increase the maximum size of memory acceptable to PHITS by redefining (increasing) the `mdas` parameter in “param.inc” in the “src” directory and then re-compiling PHITS. It may also be necessary to increase the `latmax` parameter to use huge lattice structures such as voxel phantoms. See the “4.7 Array sizes” section for further details.

Q2.3 An error occurred when I tried to use `infl:` in my PHITS input file.

A2.3 When the `infl:` command is used in the PHITS input file (e.g., “phits.inp”), it is necessary to type `file=phits.inp` in the first line of “phits.inp” file. Otherwise, another input file (e.g., “phits.in”) can be created with the first line `file=phits.inp` and used as the PHITS input file. For example:

```
phits_LinIfort < phits.in
```

See manual section 2.3.2 for further details.

Q2.4 An error occurred when I tries to execute PHITS on a Linux console, but I can execute it on Windows using the same input file.

A2.4 There are many possible causes of this error, but the most probable is a difference in the ‘return code’ used in Linux and Windows. If the input file is prepared in a Windows computer and transferred to a Linux system using FTP software, the status of the transfer mode must be checked, i.e., ‘ASCII mode’ must be selected in the FTP software.

Q2.5 Can PHITS be executed on Cygwin?

A2.5 Yes. The Cygwin option can be found in the PHITS “makefile.”

Q2.6 Which is most suitable for compiling PHITS—Intel, Fortran, or gfortran?

A2.6 We generally recommend using the Intel Fortran compiler because it can make PHITS executable files that are faster than those made using gfortran. When using gfortran, it is vital that you set '-O0' instead of using optimization options because of the possibility that an executable file with optimization will not work correctly. Thus, an executable file compiled using Intel Fortran is on average 3–5 times faster than one compiled using gfortran. Furthermore, the latest version of gfortran cannot compile PHITS (see A2.1). Note that the Intel Fortran compiler is not freeware (this is true for the Linux version as well).

Q2.7 How should the distributed memory parallel computing (MPI) or shared memory parallel computing (OpenMP) be set depending on the situation?

A2.7 Calculation using MPI is generally faster than calculation using OpenMP if the MPI protocol is installed in the computer. However, because the protocol is not generally pre-installed in Windows or Mac OS systems as a default setting, we recommend the use of OpenMP for executable files. To perform PHITS calculation on huge memories such as high energy nuclear data files or voxel phantoms, the hybrid MPI-OpenMP setting must be used to avoid the possible memory shortages that could occur if only MPI were to be used. In the hybrid setting, the number of MPI processes should be increased, but only up to the point at which the used memory does not overflow the capacity of the computer. Note that PHITS uses a number of CPU cores appropriate to the number of OpenMP processes to control the MPI processes. For example, to set the number of OpenMP processes to eight on a computer with 128 CPU cores, the maximum number of MPI processes can be specified as 128/8 – 1 = 15.

Q2.8 I got so many lost particles when I rotate a lattice structure by [transform].

A2.8 The size of outer frame surface should be smaller than the lattice structure inside it, otherwise you may observe many lost particle. Please see "\phits\lecture\advanced\voxel\" in more detail.

13.3 Questions related to Tallies

Q3.1 What is the difference between the `[t-heat]` and the `[t-deposit]` tallies?

A3.1 These two tallies essentially produce the same values of deposition energies. After version 3.05, `[t-deposit]` gives the deposition energies not only by ionization of charged particles but also kerma of neutral particles. Thus, we do not recommend to use `[t-heat]` after that version.

Q3.2 The track length or fluence of heavy ions calculated by `[t-track]` or `[t-cross]` is strange.

A3.2 This might be caused by a mis-definition of the energy mesh in the tally section. The energies of heavy ions should be defined in MeV in the tally section but should be written in MeV/u in the `[parameters]` section.

Q3.3 Results obtained by `[T-LET]` and/or `[T-SED]` tally are strange.

A3.3 The density of the material selected by `letmat` must be checked: if a material not used in your geometry is selected, it is necessary to define its absolute atomic densities (1H 6.893407e-2 16O 3.446704e-2) in the `[material]` section. In addition, if a very fine mesh (more than 10 meshes per decade) in `[t-let]` is set, the results might have zigzag structure. In that case, the LET mesh should be widened.

Q3.4 How can the statistical uncertainty be estimated from the tally output?

A3.4 In Version 2.50 and later the standard deviations or standard errors are correctly output in the tally results. See section "5.2.2 Number of history and bank" for details.

Q3.5 Can the `dump` function be used when executing PHITS in distributed memory parallel computing?

A3.5 This is possible from version 2.30 onward. Please ask the PHITS offce for more information on its detailed usage.

Q3.6 Tally results in boxes obtained using `mesh=reg` and `mesh=xyz` are inconsistent with each other.

A3.6 PHITS automatically calculates the volume of tally regions only in the cases of `mesh=xyz` or `r-z`. Thus, if `mesh=reg` is set and the `[volume]` section is not specified, the volume of the tally region is assumed to be 1 cm^3.

Q3.7 Why do some events deposit energies greater than the incident energy when using `[t-deposit]` with the `output=deposit` option.

A3.7 When exothermic nuclear reactions occur, the total energy of secondary particles becomes greater than the incident energy. Such events are also observed when `mesh=xyz` or `r-z` and `nedisp=t0` based on the output of the algorithm for considering the energy straggling of charged particles in PHITS. In such case, it is necessary to set `mesh=reg` in `[t-deposit]` and define cells corresponding to each `xyz` or `r-z` mesh in the `[cell]` section.

13.4 Questions related to source generation

Q4.1 How PHITS outputs be normalized when using an isotropic source (`s-type=9` or `10`, `dir=-all`)?

A4.1 If there is nothing inside the sphere of an 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. Therefore, to convert the tally output (/source) to the unit fluence, it is necessary to multiply the result by πr_1^2. It should be noted that, as the weight control method is employed in generating an isotropic source, event-by-event information cannot be derived from simulations using an isotropic source. To obtain event-by-event information for isotropic irradiation, set `dir=iso` in the `[source]` section (PHITS must be updated to use this function).

Q4.2 What should be done if a source particle is not created in the desired location in a cell?

A4.2 To set the source generating surface (or point) precisely on the surface of a certain cell, PHITS sometimes mis-identifies the cell where it should be located. In this case, the source surface should be moved so that it differs slightly from the cell surface.

13.5 Other questions

Q5.1 Can I use PHITS for commercial purpose?

A5.1 Basically yes. You can use PHITS for commercial purposes such as the design of radiation detector to be sold, and radiation shielding calculation for the accelerator facilities for medical use. However, you have to make an onerous contract if you want to sell a product that includes the executable file of PHITS, such as the treatment planning system based on PHITS. In that case, please contact us.

Q5.2 What kinds of physical processes cannot be handled by PHITS?

A5.2 PHITS cannot handle interactions involved with more than (or equal to) two moving particles such as particle collision experiments, transports and generations of non-ionizing radiation such as visible light, and change of chemical or material properties after irradiation. More detailed information can be found in Appendix A.

APPENDIX

A List of physical processes that cannot be handled by PHITS

- Criticality calculation
- Interaction between two (or more) moving particles (e.g. Simulation for particle collision experiments)
- Effect of electric fields generated by radiation (e.g. Laser acceleration of particles, plasma)
- Electron-induced nuclear reactions via virtual photon production
- Transport and generation of photons below 1 keV (e.g. Cherenkov light, synchrotron radiation, luminescence)
- Transport and generation of electrons below 1 keV [72]
- Physical processes related to electron current (e.g. electron avalanche)
- Charge exchange interactions [72]
- Event-by-event analysis of atomic excitation and ionization [72]
- Polarization effect [73]
- Chemical reactions (e.g. Transport and generation of radicals)
- Biological reactions (e.g. Generation of DNA damage)
- Change of chemical and material profiles after irradiation (e.g. time evolution of defects)
- Consideration of the status of a material [74] (e.g. crystal & molecular structure, temperature)
- Nuclear reactions originated from fine structure of nuclear shell [74](e.g. Li(p,n) reaction)
- Particle-induced X-ray emission (PIXE)
- Generation of delayed neutrons

[72] Feasible when the track-structure mode is used

[73] Feasible in the simulation of nuclear florescence resonance (NRF)

[74] Feasible when the nuclear data library is used

Index

PHITS *Ver. 3.17 User's Manual*

Ver.2.52	2012/12/27
Ver.2.64	2013/11/19
Ver.2.89	2017/01/11
Ver.2.90	2017/02/09
Ver.2.91	2017/02/20
Ver.2.92	2017/04/18
Ver.2.93	2017/06/16
Ver.2.94	2017/06/30
Ver.2.95	2017/07/21
Ver.2.96	2017/08/28
Ver.2.97	2017/09/21
Ver.3.00	2017/10/04
Ver.3.01	2017/10/31
Ver.3.02	2017/12/01
Ver.3.03	2018/02/01
Ver.3.04	2018/02/16
Ver.3.05	2018/03/14
Ver.3.06	2018/05/29
Ver.3.07	2018/07/05
Ver.3.08	2018/08/20
Ver.3.10	2019/04/03
Ver.3.11	2019/05/16
Ver.3.12	2019/06/20
Ver.3.13	2019/08/02
Ver.3.14	2019/08/19
Ver.3.15	2019/09/12
Ver.3.16	2019/09/26
Ver.3.17	2019/10/29

PHITS development members：
Koji Niita[1], Tatsuhiko Sato[2], Yosuke Iwamoto[2], Shintaro Hashimoto[2,†], Tatsuhiko Ogawa[2], Takuya Furuta[2], Shinichiro Abe[2], Takeshi Kai[2], Pi-En Tsai[2], Norihiro Matsuda[2], Yusuke Matsuya[2], Hunter Ratliff[2], Hiroshi Iwase[3], Nobuhiro Shigyo[4], and Lembit Sihver[5]

[1]Research Organization for Information Science & Technology (RIST)
[2]Japan Atomic Energy Agency (JAEA)
[3]High Energy Accelerator Research Organization (KEK)
[4]Kyushu University
[5]Vienna University of Technology, Austria
[†]Editors of this user's manual