PHITS & DCHAIN Tools: Python modules for parsing, organizing, and analyzing results from the PHITS radiation transport and DCHAIN activation codes

Hunter N. Ratliff [ORCID] [1,2]

1 Western Norway University of Applied Sciences, Inndalsveien 28, 5063 Bergen, Norway [ROR] **2** Japan Atomic Energy Agency, 2-4 Shirakata, Tokai, Naka, Ibaraki 319-1195, Japan [ROR]

Summary

Various areas within the nuclear sciences—such as nuclear facility design, medical physics, experimental nuclear physics, and radiation protection—rely on complex codes employing nuclear data and a large variety of physics models to simulate the transport (and interactions) of radiation through matter to answer important questions, in both research and applied contexts. For example: How should a shielding wall be designed to comply with radiation safety regulations? What dose will an individual receive in a particular exposure scenario? After how long will an irradiated radioactive sample decay enough to become safe to handle? How should an experiment be designed to make the most of limited time at an accelerator facility?

While simplified "rule of thumb" calculations can provide crude answers in some basic scenarios, fully modeling a radiation scenario is often necessary to obtain a much more precise answer. PHITS (Sato et al., 2024) (Particle and Heavy Ion Transport code System) is one such general purpose Monte Carlo particle transport simulation code, presently with over 7400 users[1]. Though PHITS can simulate a large variety of complex physics, it can only do so on the extremely short time scales of nuclear reactions. To calculate the creation and destruction of nuclides with time (activation, buildup, burnup, and decay) on any scale (seconds to centuries), distributed with and coupled to PHITS is the DCHAIN (Ratliff et al., 2020) code[2].

Within PHITS are "tallies" which score various physical quantities such as the number of particles passing through a region in space or crossing a surface, the frequency and products of nuclear interactions of various types, deposition of energy/dose, timing of interactions, displacements per atom (DPA), and more. Users provide the desired binning for the tally histograms to be created (such as specifying a range of energies of interest and how many bins the tally should have within that energy range), and the code will simulate the histories, or "lives", of many particles, outputting the aggregate distributions for the quantities being tallied, which should be converged to the "true" or "real" distributions provided a statistically sufficient number of histories were simulated (often on the order of millions or more). For a few tallies, PHITS provides the option to output "dump" files where, in addition to the histograms, detailed raw history-by-history event data are recorded to an ASCII or binary file for every history satisfying the tally's criteria and being scored by it, allowing users to, in post, create even more complex tallies and analyses than possible with the stock tallies in PHITS.

The DCHAIN code coupled to PHITS specializes in calculating nuclide inventories and derived quantities (such as activity, decay heat, decay gamma-ray emission spectra, and more) as a function of time for any arbitrary irradiation schedule from any radiation source.

---

[1] For current PHITS userbase statistics, see: https://phits.jaea.go.jp/usermap/PHITS_map_userbase.html
[2] PHITS and DCHAIN are distributed by the Japan Atomic Energy Agency and the OECD/NEA Data Bank. For more information, see: https://phits.jaea.go.jp/howtoget.html

40 The modules presented here automate the time-consuming task of extracting the numerical
41 results and metadata from PHITS/DCHAIN simulations and organizes them into a standardized
42 format, easing and expediting further practical real-world analyses. They also provide functions
43 for some of the most common analyses one may wish to perform on simulation outputs.

Statement of need

45 `PHITS Tools` and `DCHAIN Tools` serve as an interface between the plaintext (and binary)
46 outputs of the PHITS and DCHAIN codes and Python—greatly expediting further programmatic
47 analyses, comparisons, and visualization—and provide some extra analysis tools. The outputs
48 of the PHITS code are, aside from the special binary "dump" files, plaintext files formatted for
49 processing by a custom visualization code (generating Encapsulated PostScript files) shipped
50 with and automatically ran by PHITS, and those of the DCHAIN code are formatted in a variety
51 of tabular, human-readable structures. Historically, programmatic extraction and organization
52 of numerical results and metadata from both codes often required writing a bespoke processing
53 script for most individual simulations, possibly preceded by manual data extraction/isolation
54 too. `PHITS Tools` and `DCHAIN Tools` provide universal output parsers for the PHITS and
55 DCHAIN codes, capable of processing all of the relevant output files produced by each code
56 and outputting the numerical results and metadata in a consistent, standardized output format.

57 The substantial number of combinations within PHITS of geometry specification, scoring
58 axes (spatial, energy, time, angle, LET, etc.), tally types (scoring volumetric and surface
59 crossing particle fluxes, energy deposition, nuclide production, interactions, DPA, and more),
60 potential particle species, and fair amount of "exceptions" or "edge cases" related to specific
61 tallies and/or their settings highlight the utility of such a universal processing code for PHITS.
62 When parsing standard PHITS tally output, `PHITS Tools` will return a metadata dictionary, a
63 10-dimensional NumPy array universally accommodating of all possible PHITS tally output
64 containing all numerical results (structured as shown in the table below), and a Pandas
65 DataFrame containing the same numerical information, which may be more user-friendly to
66 those accustomed to working in Pandas.

axis	description
0 / `ir`	Geometry mesh: reg / x / r / tet *
1 / `iy`	Geometry mesh: 1 / y / 1
2 / `iz`	Geometry mesh: 1 / z / z *
3 / `ie`	Energy mesh: eng ([T-Deposit2] eng1)
4 / `it`	Time mesh
5 / `ia`	Angle mesh
6 / `il`	LET mesh
7 / `ip`	Particle type / group (part =)
8 / `ic`	Special: [T-Deposit2] eng2, [T-Yield] mass/charge/chart, [T-Interact] act
9 / `ierr`	= 0/1/2, Value / relative uncertainty / absolute uncertainty *

*exceptional behavior with [T-Cross] tally when `enclos = 0` is set; see full documentation

67 `PHITS Tools` is also capable of parsing the "dump" output files (both binary and ASCII formats)
68 that are available for some tallies, and it can also automatically detect, parse, and process
69 all PHITS output files within a provided directory, very convenient for PHITS simulations
70 employing multiple tallies, each with its own output file, whose output are to be further studied,
71 e.g., compared to experimental data or other simulations. The `PHITS Tools` module can be
72 used by (1) importing it as a Python module in a script and calling its functions, (2) running
73 it in the command line via its CLI with a provided PHITS output file (or directory of files) and
74 settings flags/options, or (3) running it without any arguments to launch a GUI stepping the
75 user through the various output processing options and settings within `PHITS Tools`.

76 When used as an imported module, `PHITS Tools` provides a number of supplemental functions
77 aiding with further analyses, such as tools for constructing one's own tally over the history-by-
78 history output of the "dump" files, rebinning histogrammed results to a different desired binning
79 structure, applying effective dose conversion coefficients from ICRP 116 (Petoussi-Henss et al.,
80 2010) to tallied particle fluences, or retrieving a PHITS-input-formatted [Material] section entry
81 (including its corresponding density) from a large database of over 350 materials (primarily
82 consisting of the selection of materials within the PNNL Compendium of Material Composition
83 Data for Radiation Transport Modeling (McConn et al., 2011)), among other useful functions.

84 `DCHAIN Tools` is a separate Python module for handling the outputs of the DCHAIN code
85 and is included as a submodule within the `PHITS Tools` repository. Its primary function
86 parses all of the various output files of the DCHAIN code and compiles the metadata and
87 numeric results—the confluence of the specified regions, output time steps, all nuclides and
88 their inventories (and derived quantities), and complex decay chain schemes illustrating the
89 production/destruction mechanisms for all nuclides—into a single unified dictionary object.
90 The `DCHAIN Tools` module includes some additional useful functions such as retrieving neutron
91 activation cross sections from DCHAIN's built-in nuclear data libraries, calculating flux-weighted
92 single-group activation cross sections, and visualizing and summarizing the most significant
93 nuclides (in terms of activity, decay heat, or gamma-ray dose) as a function of time. If `PHITS`
94 `Tools` is provided DCHAIN-related files, `DCHAIN Tools` will be automatically imported and its
95 primary function executed on the DCHAIN output.

96 In all, the `PHITS Tools` and `DCHAIN Tools` modules make the results produced by the PHITS
97 and DCHAIN codes far more accessible for further use, analyses, comparisons, and visualizations
98 in Python, removing the initial hurdle of parsing and organizing the raw output from these
99 codes, and provides some additional tools for easing further analyses and drawing conclusions
100 from the PHITS and DCHAIN results.

Acknowledgements

References

106 McConn, R. J., Gesh, C. J., Pagh, R. T., Rucker, R. A., & Williams, R., III. (2011). *Compendium of material composition data for radiation transport modeling*. Pacific Northwest National Lab. (PNNL), Richland, WA (United States). https://doi.org/10.2172/1023125

109 Petoussi-Henss, N., Bolch, W. E., Eckerman, K. F., Endo, A., Hertel, N., Hunt, J., Pelliccioni, M., Schlattl, H., & Zankl, M. (2010). Conversion coefficients for radiological protection quantities for external radiation exposures. *Annals of the ICRP*, *40*(2-5), 1–257. https://doi.org/10.1016/j.icrp.2011.10.001

113 Ratliff, H. N., Matsuda, N., Abe, S., Miura, T., Furuta, T., Iwamoto, Y., & Sato, T. (2020). Modernization of the DCHAIN-PHITS activation code with new features and updated data libraries. *Nuclear Instruments and Methods in Physics Research Section B: Beam Interactions with Materials and Atoms*, *484*, 29–41. https://doi.org/10.1016/j.nimb.2020.10.005

118 Sato, T., Iwamoto, Y., Hashimoto, S., Ogawa, T., Furuta, T., Abe, S.-I., Kai, T., Matsuya, Y., Matsuda, N., Hirata, Y., Sekikawa, T., Yao, L., Tsai, P.-E., Ratliff, H. N., Iwase, H., Sakaki, Y., Sugihara, K., Shigyo, N., Sihver, L., & Niita, K. (2024). Recent improvements of the particle and heavy ion transport code system – PHITS version 3.33. *Journal of Nuclear Science and Technology*, *61*(1), 127–135. https://doi.org/10.1080/00223131.2023.2275736