Joint International Conference on Supercomputing in Nuclear Applications and Monte Carlo 2013 (SNA + MC 2013)
La Cité des Sciences et de l'Industrie, Paris, France, October 27–31, 2013

Accelerated equilibrium core composition search using a new MCNP-based simulator

Jeffrey E. Seifried[1], Phillip M. Gorman[1], Jasmina L. Vujic[1], and Ehud Greenspan[1]

[1] *University of California, Berkeley, Department of Nuclear Engineering, MC 1730, Berkeley, CA 94720*

MocDown is a new Monte Carlo depletion and recycling simulator which couples neutron transport with MCNP and transmutation with ORIGEN. This modular approach to depletion allows for flexible operation by incorporating the accelerated progression of a complex fuel processing scheme towards equilibrium and by allowing for the online coupling of thermo-fluids feedback. MocDown also accounts for the variation of decay heat with fuel isotopics evolution. In typical cases, MocDown requires just over a day to find the equilibrium core composition for a multi-recycling fuel cycle, with a self-consistent thermo-fluids solution–a task that required between one and two weeks using previous Monte Carlo-based approaches.

KEYWORDS: Monte Carlo neutron transport, depletion, recycling, parallel, decay heat, conversion, thermo-fluids, coupled neutronics thermo-fluids

I. Introduction

Existing core simulation codes are either insufficiently accurate or computationally inefficient in the search for the equilibrium composition of recently proposed reduced-moderation boiling water reactor designs.[1,2] The cores of such reactors feature a strongly varying axial coolant density distribution, hard and axially varying neutron spectra, and large axially varying flux gradients, which together mandate the use of continuous-energy three-dimensional Monte Carlo neutron transport. Tight physical coupling between the spatial variation of the fission power density and coolant density require tight numerical coupling between neutron transport and thermo-fluids models. Interest in, primarily, the equilibrium cycle necessitates an efficient means of finding the equilibrium core composition.

This paper describes a new tool named MocDown which meets these and other needs. In section II, the general design and programming approaches which were taken in developing MocDown are discussed. Then, in section III, the RBWR-Th core design, which serves for numerical illustrations of MocDown's utility, is briefly described. Next, section IV describes and demonstrates the accelerated recycling scheme that MocDown takes in seeking equilibrium core compositions for multi-recycling fuel cycles. Section V shows an example of the online thermo-fluids coupling. Finally, in section VI, MocDown's simple approach for source rate scaling, which accounts for isotopic composition-dependent decay heat, is described.

II. MocDown general programming characteristics

MocDown is an advanced Monte Carlo depletion simulator. Just as MOCUP,[3] MONTEBURNS,[4] IMOCUP,[5] Mocup.py,[6] VESTA,[7] and countless other codes do, MocDown simulates the depletion of nuclear reactor cores by coupling neutron transport with MCNP[8] and transmutation with ORIGEN2.2.[9] In addition, it facilitates the search for the equilibrium composition of multi-recycling fuel cycles in an efficient manner, enables online coupling of thermo-fluids models, and employs a simple approach towards neutron source rate scaling. MocDown also incorporates many other programming best practices which provide for a robust, reliable experience for users.

MocDown is written in object-oriented Python 3. Auxiliary operations, like thermo-fluids models and fuel processing, are completely customizable in external modules. These modules take advantage of interface methods (e.g., `GetBurnCells` or `GetIsDecayStep`), which pass all data in memory and eliminate error prone I/O and file parsing. For example, a Python 3 library, which offers IAPWS-IF97 steam table property lookups within Python,[10] was readily found and integrated into a simple thermo-fluids model. This modular approach also allows MocDown to remain separate and intact for a number of projects, greatly simplifying version control and software verification.

Execution of ORIGEN2.2 is concurrently threaded using standard Python 3 libraries, enabling the depletion of twenty regions in parallel (although this number depends upon the hardware, it is thought to be a typical number). When depleting large systems in parallel, runtime speedups of 6-7× have been observed over serial execution.

Transmutation constants (region-wise total flux magnitudes and region-, isotope-, and reaction-wise one-group cross-sections) are extracted with a single MCNP tally which is dynamically generated. This removes restraints upon the number of regions or isotopes that can be depleted. In contrast, MONTEBURNS and MOCUP use one tally per depletion region and can tolerate only a certain number of regions, based upon their configuration and the version of MCNP. The isotopes

whose transmutation constants are calculated with MCNP are determined according to the isotope's contribution to regional molar and mass densities and absorption and fission rates. This removes the need for a priori specification of which isotopes are tracked and automatically allows different sets of isotopes to be tracked in different depletion regions. MONTEBURNS incorporates this strategy, but requires a master list of isotopes to be provided.

Regular expressions parsing is used for robust extraction of transmutation constants from MCNP and isotopic inventories from ORIGEN2.2 output files. This allows for the flexibility to use MCNP5, MCNPX, or MCNP6 for neutron transport calculations and removes restrictions on the formatting of neutron transport code input files. Other codes are confined to certain versions of MCNP due to periodic changes in the formatting of output files. For simulations which deplete many regions and track many isotopes, regular expressions can also parse text faster than finite state-based parsing approaches.

Depletion control parameters (e.g., depletion power, cycle length, the list of cells to deplete, etc.) are defined in a verbose, free-format input file. This centralization of inputs to a single location eliminates duplication and removes any requirements for special flags or comments within the neutron transport input file.

MocDown periodically serializes, compresses, and writes objects which are in memory to hierarchical data files. This archival memory dump facilitates simple post-processing and debugging of simulation results through MocDown's object-oriented interfaces. Additionally, these serializations offer a convenient means to reload depletion control and state parameters into memory in order to restart a simulation which may have ended prematurely.

III. Description of the RBWR-Th unit cell model

The RBWR-Th is an under-moderated boiling water reactor, whose hard spectrum allows its unenriched charge to breed fuel self-sufficiently while operating within the ABWR pressure vessel.[2] The core design is an iteration of the RBWR-AC, which conforms to the same constraints,[1] but which departs from the RBWR-AC in several ways: thorium is used as the fertile fuel instead of depleted uranium, the internal blanket is eliminated, absorbers in the upper and lower reflectors are eliminated, and the fissile region is elongated.

A single pin unit cell with 55 depletion regions is used for neutron transport and a one dimensional two-phase heat balance and void fraction model is used for thermo-fluids feedback. Each 0.855 cm OD fuel pin is axially apportioned into a 50 cm long lower blanket, a 111 cm long central seed, and a 70 cm long upper blanket, and is arranged in a hexagonal lattice of pitch 1.135 cm. The fuel is assumed to be 93% of the theoretical density and is clad with 0.06 cm thick zirconium. At the beginning of each cycle, the blankets are charged with natural thoria and the seed is charged with the processed discharge from the previous cycle. After being depleted at 20 kWth for 2287.5 EFPD and then cooling for three years, the fuel is discharged and processed. In this recycling process shown in Figure 1, fission products are discarded; all transthoria are harvested from the blankets and seed, and recharged into the seed of the next cycle; and the balance is made up with natural thoria.

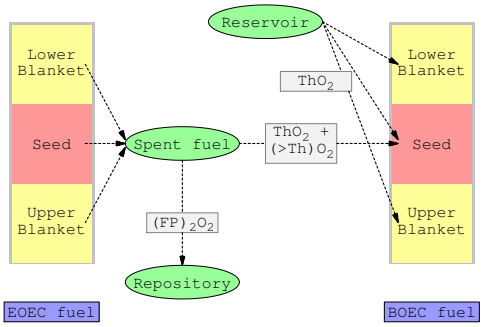

Figure 1: The RBWR-Th fuel processing scheme discharges only fission products, incinerates all heavy metals, and requires no enrichment.

IV. Accelerated recycling scheme for the search of equilibrium core compositions

MocDown is most powerful when modeling a system with a multi-recycling fuel cycle (whose charge is made up of a portion of its own discharge) while simultaneously searching for the equilibrium (the point after which the core composition of cycles remains unchanged). When is it used for this purpose, it is beneficial to consider its operation as the self-consistent convergence of three inter-dependent models: neutron transport, transmutation and recycling, and thermo-fluids. Transport derives power distributions (P_{th}) and transmutation constants (σ,ϕ) from thermo-fluids conditions and isotopic inventories, transmutation and recycling derives isotopic inventories (N) from transmutation constants, and thermo-fluids derive thermo-fluids conditions (ρ,T) from power distributions. These dependencies are illustrated in Figure 2.

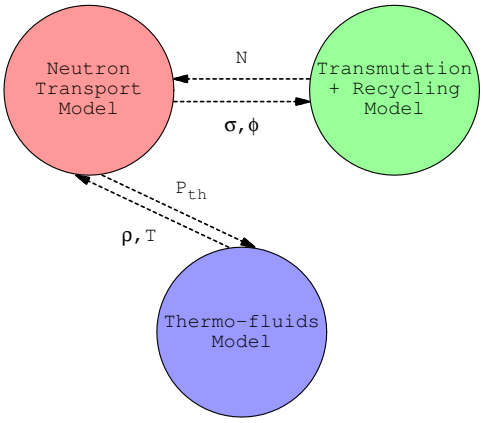

Figure 2: Thermo-fluids-coupled depletion simulations can be broken into three independent models which must be self-consistent: neutron transport, transmutation and recycling, and thermo-fluids.

Traditionally, depletion (without thermo-fluids feedback) is performed by alternating between neutron transport and transmutation in lockstep, holding N constant during the former and holding σ and ϕ constant during the latter. This ensures a good degree of consistency between the two models. A typical cycle

requires 20 time-steps. When thermo-fluids is coupled, each single neutron transport execution is replaced by a fixed point iteration between neutron transport and thermo-fluids models, a process which continues until agreement is reached. This doubles the number of neutron transport executions per cycle on the average. If the cycle equilibrium is sought, dozens of cycles will need to be simulated. The result is an approach with requires on the order of one thousand executions of the neutron transport and thermo-fluids models and one half as many executions of the transmutation model.

The alternative approach taken by MocDown recognizes the >20× imbalance of computational expense between the neutron transport/thermo-fluids couplet and the transmutation model, which is only exaggerated by concurrently threaded transmutation. By loosening the numerical coupling of the three models somewhat, the number of neutron transport/thermo-fluids couplets can be drastically reduced. While this approach slightly increases the number of cycles required for convergence, the overall runtime plummets.

This accelerated approach to recycling is depicted in Figure 3 and proceeds as follows. The outer loop performs full-fidelity cycles (with neutron transport-updated transmutation constants and thermo-fluids feedback) until some norm of successive cycle multiplication factors falls below a tolerance. Following each failed (i.e., unconverged) iteration of the outer loop, the scheme enters the inner loop. The inner loop performs accelerated cycles (transmutation constants are held constant, avoiding the neutron transport/thermo-fluids couplet), pushing the fuel through depletion and recycling until some norm of successive cycle isotopic abundances falls below a tolerance. Upon success (i.e., convergence) of the outer loop, the cycle equilibrium is declared to be found and the search is complete.

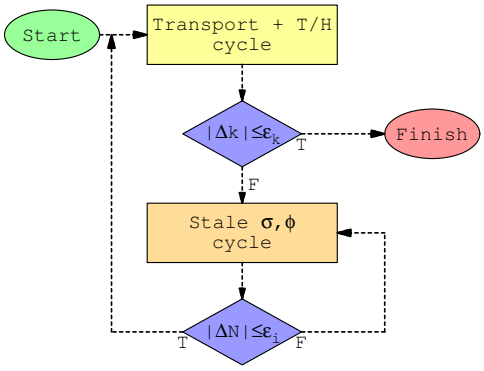

Figure 3: The accelerated recycling scheme used in MocDown eliminates many executions of neutron transport and thermo-fluids calculations by holding transmutation constants fixed in the search for equilibrium fuel cycles.

In order to demonstrate the effectiveness and efficiency of MocDown's accelerated recycling scheme, the analysis is performed and discussed for the RBWR-Th core design described in section III. As a single full-fidelity cycle for this design completes in 4.5 hours, 40 cycles (which would be required for convergence towards cycle equilibrium) would take over a week to finish. However, when MocDown is used for the equilibrium search, most of the full-fidelity cycles are replaced with accelerated cycles which complete in 12 minutes. In total, 4 full-fidelity and 85 accelerated cycles are simulated and the cycle equilibrium is found in only 35 hours. This is almost ten times faster than the traditional approach.

Convergence after this number of accelerated cycles is demonstrated qualitatively with Figure 4, which shows that the core composition ceases to change with additional cycles. Full-fidelity cycles are performed at the dashed line and accelerated cycles are performed in between. Following each full-fidelity cycle, transmutation constants are fixed and the isotopic abundances asymptotically approach a new equilibrium; each new equilibrium differs less and less from the previous one. Figure 5 qualitatively shows that the multiplication factor

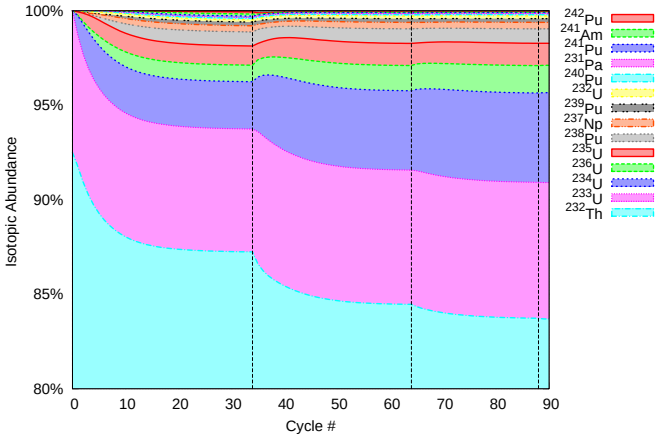

Figure 4: The MocDown accelerated recycling scheme efficiently finds the equilibrium cycle, whose isotopic composition matches that of its successor. Dashed lines denote full-fidelity cycles.

for successive full-fidelity cycles has converged. A quantitative

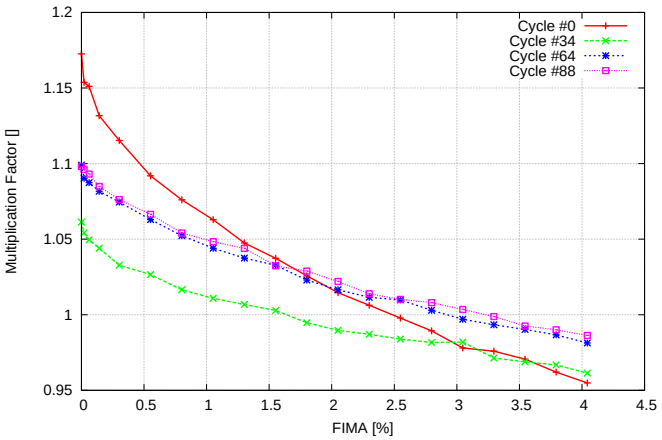

Figure 5: The MocDown accelerated recycling scheme efficiently finds the equilibrium cycle, whose cycle multiplication factor matches that of its successor.

measure of convergence is demonstrated in Figure 6, which shows the ∞-norm of isotopic abundance differences versus cycle.

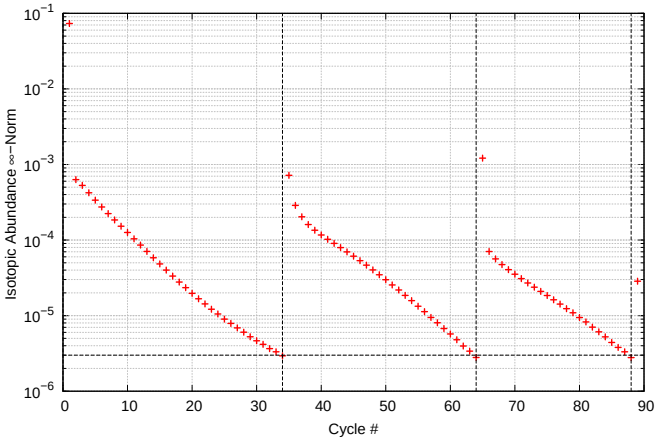

Figure 6: The difference in isotopic composition abundances becomes progressively smaller upon each accelerated cycle.

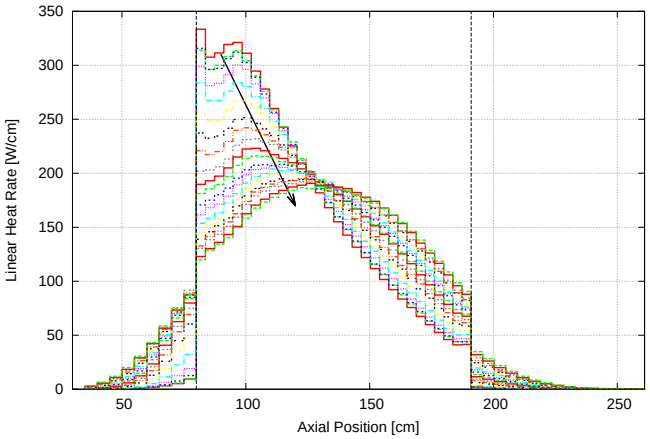

Figure 7: As the system accrues burnup, the linear heat generation rate traverse tends to flatten towards a chopped-cosine distribution.

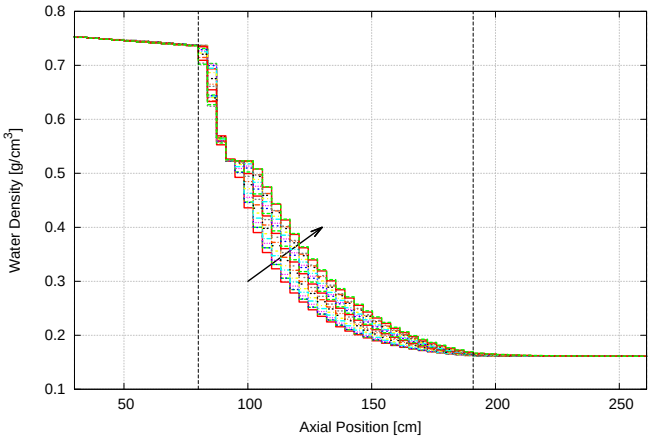

Figure 8: The corresponding coolant density traverse drifts upwards with the linear generation rate.

V. Online thermo-fluids coupling

As described earlier, online thermo-fluids coupling is achieved by performing a fixed-point iteration between neutron transport and thermo-fluids models until self-consistency is reached. MocDown minimizes the number of these iterations by using the most relevant thermo-fluids results as a guess for the converged values–those of the previous depletion step for the first full-fidelity cycle and those of the same depletion step of the previous cycles for the following full-fidelity cycles.

During parametric studies of the RBWR-Th core, several design variants were found whose neutron transport/thermo-fluids couplets oscillated between multiple solutions and then converged slowly to self-consistency in an underdamped fashion. For these cases, MocDown allows a relaxation factor to be applied to the thermo-fluids results. When under-relaxation is used, the most recent results are effectively averaged with the previous results and convergence occurs in three or four instead of tens of iterations.

The RBWR-Th model is once again used to demonstrate the necessity and effectiveness of thermo-fluids coupling in MocDown. For this design, the thermo-fluids model is composed of a one-dimensional channel model which: (1) extracts the thermal power axial traverse from MocDown; (2) calculates the coolant quality traverse according to steam properties and coolant flow parameters; (3) estimates the void fraction using one of a number of drift-flux correlations; (4) calculates the coolant density traverse; (5) determines convergence by comparing successive coolant density traverses; and (6) performs a number of secondary operations, including calculating the minimum critical power ratio and pressure drop. Figure 7 shows how the linear heat generation rate traverse shifts upwards over a cycle, as fissile material is consumed and bred. As a consequence, the coolant density traverse also drifts upwards as shown in Figure 8.

VI. Neutron source rate scaling and decay heat

Depletion analysis requires accurate knowledge of the burnup-dependent neutron flux magnitude. In Monte Carlo neutron transport codes, quantities like the flux magnitude are reported on a per source (fission) neutron basis. Consequently, in order to obtain the flux magnitude on a per second basis, the number of source neutrons generated per second (S) must be known. Only three quantities are required to deduce S: the total energy deposited in the core per source neutron (E), the thermal power at which the core operates (P_{th}), and the decay heat (P_d). E is made up of neutron and photon heating which includes fission, radiative capture, Compton scattering, and other endothermic and exothermic reactions. P_{th} is the sum of particle heating (the product of E and S) and P_d. Putting all of this together, an expression can be written for S:

$$S = \frac{P_{th} - P_d}{E}. \quad (1)$$

In this formulation, each term is known or can be derived quite easily: P_{th} is specified for each simulation; E can be estimated directly with MCNP using a single $F6:np$ neutron/photon track-length estimated energy deposition tally [11, 2-88]; and P_d can be calculated with isotopic inventories, half-lives, and recoverable energy Q-values.

With this neutron source rate, heating rates can be derived which are self-consistent and which appropriately account for decay heat. Figure 9 demonstrates how the decay heat starts

at 0% and then quickly saturates to around 5.5% of the total thermal power and consequently how the neutron source rate drops quickly to 93% of its initial value. It is important to note that if decay heat is neglected, flux magnitudes and nuclear reaction rates will be in error by +5-7%.

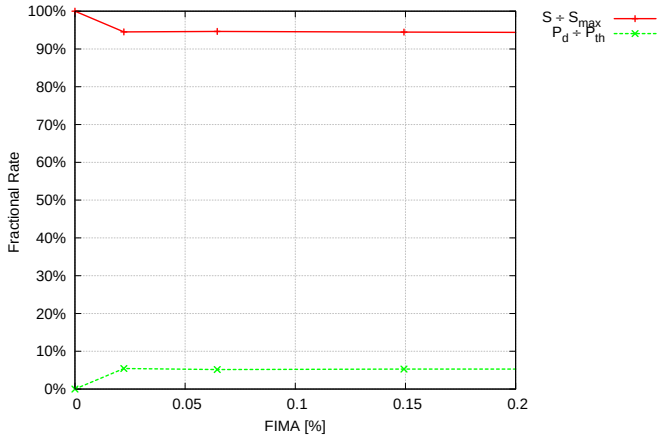

Figure 9: Because the thermal power is held constant, the neutron source rate, which is used to scale all nuclear reaction rates, depends upon the decay heat from radioisotope inventories.

Another point which deserves mentioning is the matching of flux magnitudes between MCNP and ORIGEN2.2. When ORIGEN2.2 is instructed to transmute at a constant power, it internally calculates a flux magnitude based upon that provided power, isotopic fission rates, and isotopic recoverable fission energies ($Q_{Z,A}$). ORIGEN2.2 calculates $Q_{Z,A}$ using the correlation:

$$Q_{Z,A}\,[\text{MeV}] = 1.29927 \times 10^{-3} Z^2 \sqrt{A} + 33.12, \qquad (2)$$

where Z and A are the atomic and mass numbers respectively of a fissile isotope. In order to ensure that this flux magnitude which ORIGEN2.2 calculates is consistent with MCNP, MocDown provides ORIGEN2.2 with a special "ORIGEN power" calculated from $Q_{Z,A}$ in Equation 2. This ORIGEN power differs from all other physical powers and exists only to ensure consistency in neutron transport and transmutation fluxes and is otherwise not used.

VII. Conclusions

MocDown is a modern Monte Carlo depletion and recycling simulator which incorporates many programming best practices and offers a robust user experience. Compared to MOCUP, our previous depletion and recycling workhorse, MocDown automates tasks like isotope tracking and tally generation, allows usage of current versions of MCNP, centralizes inputs to a single location, simplifies post-processing, and greatly accelerates transmutation. The modular approach towards online coupling of thermo-fluids presents an easy and flexible way of simulating a nuclear system with self-consistency. The accelerated recycling scheme (with concurrent threading of transmutation) efficiently finds the equilibrium core composition of multi-recycling fuel cycles. The simple approach for source rate scaling accurately accounts for decay heat and requires very little computational overhead beyond standard transmutation constant estimation.

Acknowledgments

This research was performed using funding received from the U.S. Department of Energy Office of Nuclear Energy's Nuclear Energy University Programs. This research is based upon work partially supported by the U.S. Department of Energy National Nuclear Security Administration under Award Number DE-NA0000979.

References

1) R. Takeda, J. Miwa, and K. Moriya, "BWRs for Long-Term Energy Supply and for Fissioning Almost All Transuranics," *Proc. Proceedings of the 2007 Advanced Nuclear Fuel Cycles and Systems (Global) Conference*, September, 2007.
2) F. Ganda, J. Vujic, and E. Greenspan, "Thorium self sustaining BWR cores," *Proc. Proceedings of the 2011 International Congress on Advances in Nuclear Power Plants (ICAPP) Embedded Topical Conference*, May, 2011.
3) R. Moore, B. Schnitzler, C. Wemple, R. Babcock, and D. Wessol, "MOCUP: MCNP-ORIGEN2 Coupled Utility Program," INEL-95/0523, INL (1995).
4) H. Trellue and D. Poston, "MONTEBURNS 2.0 | An Automated Multi-Step Monte Carlo Burnup Code System," PSR-455, ORNL (2003).
5) F. Heidet, "IMOCUP Description," internal memo, UCBNE (2010).
6) A. Cisneros, "mocup.py," internal memo, UCBNE (2012).
7) W. Haeck, "VESTA User's Manual - Version 2.1.0," DSU/SEC/T/2011-81, IRSN (2012).
8) J. Goorley et al., "Initial MCNP6 Release Overview - MCNP6 version 1.0," LA-UR-13-22934, LANL (2013).
9) A. Croff, "A User's Manual for the ORIGEN2 Computer Code," TM-7175, ORNL (1980).
10) jjgomera, "iapws 1.0.0," 2012, http://pypi.python.org/pypi/iapws/.
11) X-5 Monte Carlo Team, "MCNP – A General Monte Carlo N-Particle Transport Code, Version 5 | Volume I: Overview and Transport," LA-UR-03-1987, LANL (2008).