# Standardized Verification of the Cyclus Fuel Cycle Simulator

Jin Whan Bae[1], Joshua L. Peterson-Droogh[2], Kathryn D. Huff[1]

[1] *Dept. of Nuclear, Plasma, and Radiological Engineering, University of Illinois at Urbana-Champaign, Urbana, IL*
[2] *Oak Ridge National Laboratory, Oak Ridge, TN*

---

## Abstract

Many nuclear fuel cycle simulators can analyze transitions from once-through to advanced nuclear fuel cycles. Verification studies compare various fuel cycle analysis tools to test agreement and identify sources of difference. A recent verification study, B. Feng et al., "Standardized verification of fuel cycle modeling," Annals of Nuclear Energy, vol. 94, pp. 300-312, Aug. 2016 [1] established transition scenario test case specifications and accordingly evaluated national laboratory nuclear fuel cycle simulators, DYMOND, VISION, ORION, and MARKAL. This work verifies the performance of Cyclus, the agent-based, open-source fuel cycle simulator, using the test case specifications in Feng et. al. In this work, Cyclus demonstrates agreement with the results from the previous verification study. Minor differences reflect intentional, detailed material tracking in the Cycamore reactor module. These results extend the example results in Feng et. al to further enable future verification of additional nuclear fuel cycle simulation tools.

---

## 1. Introduction

Fuel cycle simulators guide and inform Nuclear Fuel Cycle (NFC) research directions and policy choices. Various institutions have developed fuel cycle simulators targeted at their unique needs, using various methods and structures to simulate material flow in the NFC. Algorithmic differences make validation

*Preprint submitted to Annals of Nuclear Energy* *September 15, 2018*

studies necessary to establish confidence in software capabilities and agreement amongst analysis tools.

A previous verification study [1] compared four well-known NFC simulators DYMOND [2], VISION [3], ORION [4], and MARKAL [5]. The results from each simulation tool were compared to a set of 'model solutions' that were generated from an excel worksheet for different metrics (e.g. fuel loading in reactor, Used Nuclear Fuel (UNF) inventory) in a transition scenario, and showed excellent agreement with the result from the Feng et. al [1].

The present work benchmarks the CYCLUS nuclear fuel cycle simulator [6] against the verification study results from Feng et. al [1]. Using the scenario definitions established in that validation study, this work uses CYCLUS and its additional modules library, CYCAMORE, to simulate the test case: a transition scenario from an open fuel cycle to an advanced fuel cycle with reprocessing. Comparison between these results and the Feng et. al 'model solutions' show excellent agreement.

### *1.1.* CYCLUS

CYCLUS is an *agent-based* fuel cycle simulation framework [6], meaning that each reactor, reprocessing plant, fuel fabrication plant, and other fuel cycle facility is modeled as a discrete entity. A CYCLUS simulation contains prototypes, fuel cycle facilities with pre-defined parameters, that are deployed as `Facility` agents. `Institution` agents in CYCLUS deploy or decommission `Facility` agents which `Region` agents, in turn, manage the `Institution`s. The CYCAMORE library contains customized `Facility`, `Institution`, and `Region` models which vary in complexity and purpose [7]. The CYCAMORE `DeployInst` is used as the `Institution` archetype for this work, since it deploys facilities at user-defined timesteps.

## 2. Methodology

Feng et al. comprehensively defines simulation parameters sufficient to reproduce the transition scenario in CYCLUS. In this study, we used the CYCAMORE

[6] archetype library to model all fuel cycle facilities. CYCAMORE libraries contain simple fuel cycle facility models.

CYCLUS results are output in either `.sqlite` or `.h5` format. In this study, we used the `.sqlite` format and analyzed the results using a python script. The post-processed output data, was overlapped with the results with the model solution from the verification study [1]. The input file and analysis procedures are all in [8]. The analysis and benchmark were performed iteratively, where we improve the original result by communicating with the authors of the benchmark. We analyzed the reasons for the differences from the original result, and made small edits in the source code. Major differences in the facility behavior algorithms were not edited but simply explained in detail as to how they contributed to the difference in the results.

## 3. Fundamental Modeling Differences in Cyclus

CYCLUS has fundamental algorithmic differences from the fuel cycle analysis tools used in the benchmark [1].

CYCLUS has a default time step of a month. The verification study solutions are evaluated with 1-year time steps, so cumulative and annual averages were used. For example, decommissioning of facilities occurs at the end of a timestep, and building of facilities occurs at the beginning of a timestep.

The CYCAMORE recipe reactor depletes half of its core when decommissioned, whereas the software tools in the benchmark [1] deplete all their reactors' fuel when decommissioned. This causes a major discrepancy for the transuranic elements (TRU) inventory. For this study, we changed the CYCAMORE source code to deplete all its assemblies to the depleted recipe. Also, the CYCAMORE recipe reactor treats each batch (and assembly) as a discrete material, while some tools assume continuous fuel discharge. This produces differences in the results because the batches in the benchmark [1] are in time-averaged values. In this study, the Light Water Reactor (LWR) batch size and cycle time is increased, while decreasing the batch number to keep the core size constant. We round up the Sodium-Cooled Fast Reactor (SFR) batch number, while the batch size and cycle time are kept constant. This increases the core size by 1.08%, which is negligible, but will be discussed in the results section. The differences are listed in table 1.

Table 1: Difference in Batch number and core size

| **Category** | **Model Solution [1]** | **Cyclus** |
|---|---|---|
| LWR Batches | 4.5 | 3 |
| LWR Batch size [tHM] | 19.91 | 29.86 |
| LWR Core size [tHM] | 89.59 | 89.59 |
| LWR Cycle time | 1 year | 1.5 years |
| SFR Batches | 3.96 | 4 |
| SFR Batch size [tHM] | 3.95 | 3.95 |
| SFR Core size [tHM] | 15.63 | 15.8 |

Note that all these differences could have been resolved by changing the archetype source code. However, the only change made was the reactor depletion behavior at decommission due to its large impact on plutonium inventory. The goal of this study is to show current Cyclus agreement with other simulators and identify differences, not to alter Cyclus to match the other tools.

## 4. Results

We represent each Cyclus result as a solid line, and the benchmark solution as a dotted line for visualization. The results are simply a reproduction of the plots displayed in the benchmark. We obtained the benchmark solutions through personal contact with benchmark author Bo Feng at Argonne National Laboratory.

Figure 1 shows the deployed reactor capacity, and figure 2 shows the LWR retirement and SFR deployment. The two plots show exact agreement with the benchmark solutions.

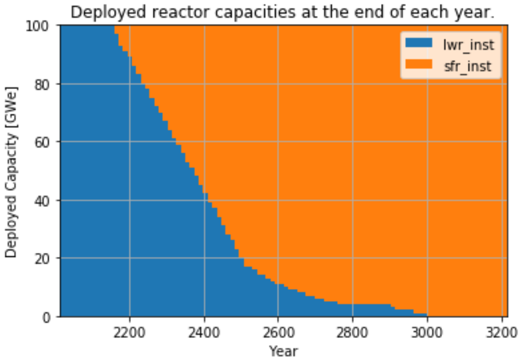

Figure 1: Deployed reactor capacities at the end of each year.

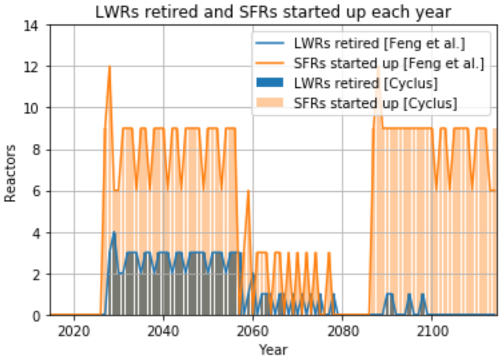

Figure 2: LWRs retired and SFRs started up each year.

Figure 3 shows the annual fuel loading rate. The initial fuel loading for 100 LWR reactors was edited to match the plot in the verification study results. The oscillations caused by the 18 month refueling period were aggregated into 12 month groups. As a result the total fuel loaded are equal for both plots.

Although indistinguishable in figure 3, there is a small difference between SFR fuel loading proportional to the core mass difference, as mentioned in the previous section. Figure 4 shows the differences normalized by the core mass differences, overlapped with the SFR deployment. This shows that the differences only occur during deployment due to the difference in core mass.

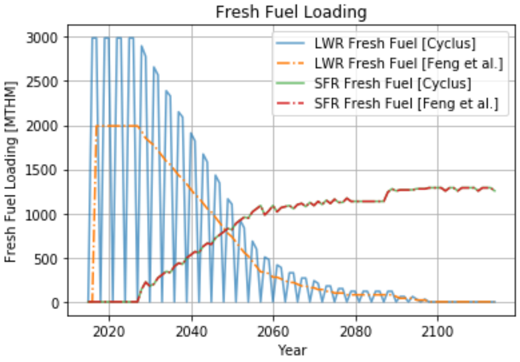

Figure 3: Annual fresh fuel loading rates (first cores and reload fuel).

Figure 5 shows the inventory of discharged UNF in the mandatory cooling stage (four years for LWR, one year for SFR). It also oscillates between the benchmark's solution and converges, caused by the influx and the outflux of UNF into and out of the storage facility. The SFR inventory and fuel loading solutions exactly matches the benchmark solutions, minus the small (1.07%) difference due to core size.

Figure 6 shows the amount of cooled UNF waiting for reprocessing. The value is calculated by subtracting the cumulative difference between the cooled inventory and the UNF reprocessing throughput. The oscillation is between the cooled inventory in the storage facility before (high) and after (low) it sends its inventory for reprocessing.

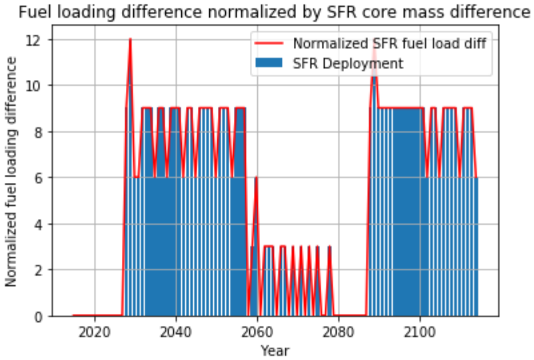

Figure 4: Difference between annual fresh SFR fuel loading rates (Cyclus - Benchmark) normalized by the core mass difference of an SFR due to fractional batch size.

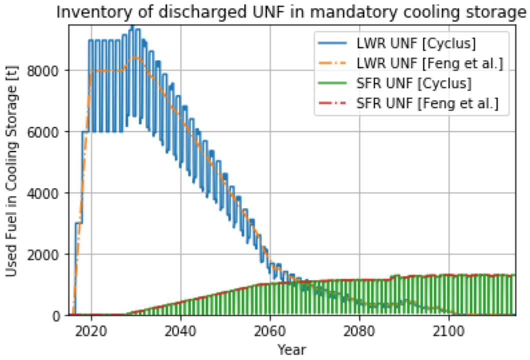

Figure 5: Inventory of discharged UNF in mandatory cooling storage.

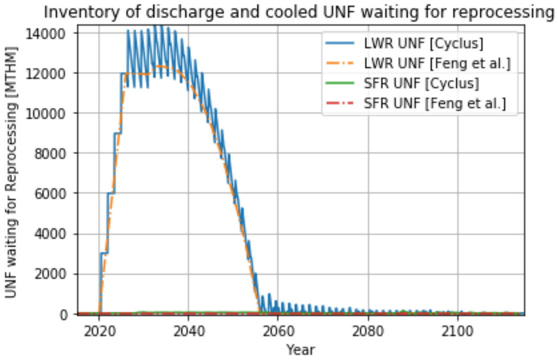

Figure 6: Inventory of discharged and cooled UNF waiting for reprocessing.

Figure 7 shows the reprocessing throughput, which oscillates around the benchmark solution. No oscillation exists in the beginning because the LWR UNF reprocessing plant throughput peaks at 2,000 tons per year.

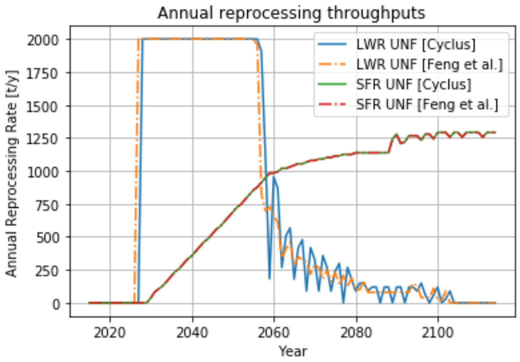

Figure 7: Annual reprocessing throughputs.

Figure 8 shows the inventory of unused TRU recovered from UNF. The Cyclus results follow the benchmark solutions closely. However, the larger SFR core size causes Cyclus results to be smaller than the benchmark results, since more TRU is used to start up the newly deployed SFRs. The difference decreases as the SFRs decommission, discharging more UNF (and hence TRU) than the benchmark.

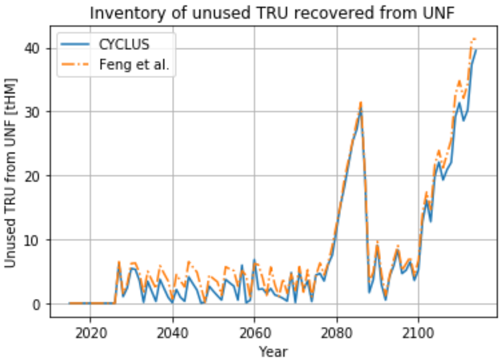

Figure 8: Inventory of unused TRU recovered from UNF.

## 5. Discussion

We benchmarked Cyclus with results from an established verification study and saw good agreement in a transition scenario.

Throughout this work, two major differences were identified that led to the deviation of Cyclus results from that of the benchmark solution. First, the Cycamore reactor depletes only half of its core when decommissioned. Second, Cyclus, unlike other codes examined in the benchmark (except ORION), fully resolves discrete batches for fuel discharge. We resolve the first discrepancy by changing one line in the source code.

This study proves Cyclus as a capable tool for modeling fuel cycle transition scenarios, and shows promise for expansion and future development.

## 6. Acknowledgments

This work was supported by the Nuclear Engineering Science Laboratory Synthesis (NESLS) program as well as the DOE Office of Nuclear Energy's Nuclear Energy University Program (Project 16-10512) 'Demand-Driven Cycamore Archetypes'. We thank Eva Davidson from Oak Ridge National Laboratory (ORNL) and Bo Feng from Argonne National Laboratory (ANL) for their aid in providing benchmark solutions and insight for this work. Additionally, the authors are thankful for the thoughtful reviews from our colleagues in the Advanced Reactors and Fuel Cycles research group.

Prof. Huff is supported by the Nuclear Regulatory Commission Faculty Development Program, the Blue Waters sustained-petascale computing project supported by the National Science Foundation (awards OCI-0725070 and ACI-1238993) and the state of Illinois, the NNSA Office of Defense Nuclear Nonproliferation R&D through the Consortium for Verfication Technologies and the Consortium for Nonproliferation Enabling Capabilities (awards DE-NA0002576 and DE-NA0002534), and the International Institute for Carbon Neutral Energy Research (WPI-I2CNER), sponsored by the Japanese Ministry of Education, Culture, Sports, Science and Technology.

## References

[1] B. Feng, B. Dixon, E. Sunny, A. Cuadra, J. Jacobson, N. R. Brown, J. Powers, A. Worrall, S. Passerini, R. Gregg, Standardized verification of fuel cycle modeling, Annals of Nuclear Energy 94 (2016) 300–312. doi:10.1016/j.anucene.2016.03.002.
URL `http://www.sciencedirect.com/science/article/pii/S0306454916301098`

[2] A. M. Yacout, J. J. Jacobson, G. E. Matthern, S. J. Piet, A. Moisseytsev, Modeling the Nuclear Fuel Cycle, in: The 23rd International Conference of the System Dynamics Society," Boston, Citeseer, 2005.
URL `http://www.inl.gov/technicalpublications/Documents/3169906.pdf`

[3] J. Jacobson, A. Yacout, G. Matthern, S. Piet, D. Shropshire, R. Jeffers, T. Schweitzer, Verifiable fuel cycle simulation model (VISION): a tool for analyzing nuclear fuel cycle futures, Nuclear Technology 172 (2) (2010) 157–178.

[4] R. Gregg, C. Grove, Analysis of the UK Nuclear Fission Roadmap using the

ORION fuel cycle modelling code, in: Proc of the IChemE nuclear fuel cycle conference, Manchester, United Kingdom, 2012.

[5] C. Shay, J. DeCarolis, D. Loughlin, C. Gage, S. Yeh, S. Vijay, E. L. Wright, EPA US national MARKAL database: database documentation, US Environmental Protection Agency, Office of Research and Development.

[6] K. D. Huff, M. J. Gidden, R. W. Carlsen, R. R. Flanagan, M. B. McGarry, A. C. Opotowsky, E. A. Schneider, A. M. Scopatz, P. P. H. Wilson, Fundamental concepts in the Cyclus nuclear fuel cycle simulation framework, Advances in Engineering Software 94 (2016) 46–59, arXiv: 1509.03604. doi:10.1016/j.advengsoft.2016.01.014.
URL `http://www.sciencedirect.com/science/article/pii/S0965997816300229`

[7] K. D. Huff, M. Fratoni, H. R. Greenberg, Extensions to the cyclus ecosystem in support of market-driven transition capability, Tech. rep., Lawrence Livermore National Laboratory (LLNL), Livermore, CA (2014).

[8] J. W. Bae, G. Park, G. Chee, K. Huff, T. Kennelly, P. Speaks, P. Wilson, A. Scopatz, arfc/transition-scenarios: Standardized Verification of the Cyclus Fuel Cycle Simulator, Zenodo GitHub. doi:10.5281/zenodo.1419110.
URL `https://doi.org/10.5281/zenodo.1419110`