IMPLEMENTATION AND VALIDATION OF OPENMC IN SALTPROC

BY

OLEKSANDR REDIN YARDAS

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Nuclear, Plasma, Radiological Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2022

Urbana, Illinois

Master's Committee:

Research Scientist Madicken Munk, Advisor
...

Abstract

Abstract.

Acknowledgments

Acks.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

Increasing Greenhouse gas (GHG) emissions due to human activity since the Industrial Revolution drives observed increases in average global surface temperature via the greenhouse effect[22] [24] (**global warming**). The long term impacts of global warming... Rapid decrease in global GHG emissions can dampen the impact of global warming. Transitioning from fossil-fuel-based electricity production methods to zero-emission [1] electricity production methods (**decarbonization**) would contribute significantly to a global GHG emission reduction, and in turn dampen the effects of global warming. Our three main options for decarbonization are renewables (solar, wind, geothermal), hydropower, and nuclear power. We will need all of these technologies working together if we want to successfully decarbonize, however each technology has a specific use case where it works best; solar and wind are best suited for storing up energy for later use on a large scale[2] which can then offset rapid fluctuations in electricity usage, whereas nuclear, hydropower, and geothermal are better suited to providing a base-load power[2]. Geothermal and hydropower are limited by geographical features (geothermal activity and elevated water sources), making nuclear the most flexible option[3] for a future carbon-free base-load. Therefore, we should maintain and develop nuclear power technologies so that they can contribute to decarbonization efforts[4].

1.2 Current and future trajectories of nuclear power

Nuclear power currently constitutes a fifth of US domestic electricity production, and half of US zero-carbon electricity production [3] [1]. Unfortunately, our current nuclear fleet faces several threats to its long-term survival. The most relevant of these is the state of the electricity market: record-low natural gas prices and

[1] zero-emissions during operation; there are life-cycle carbon costs associated with any form of power production

[2] this is a product of the *intermittency* of wind and solar; solar intensity and wind speed are highly variable and unpredictable long term

[3] provided a suitable coolant exists

[4] despite these positive qualities, nuclear power is a divisive technology with legitimate technological and policy concerns. Fortunately, many of these concerns have technological and procedural solutions

increases in subsidies for renewables with little similar compensation for nuclear make it uncompetitive in the current market and unsustainable in the long term if current conditions continue[5] [37].

Assuming that we resolve the economic threats to our current nuclear fleet, they are – like all things – subject to aging and deterioration; at some point in the future, we will need to shut down and decommission them. As explained in the previous section, nuclear power will be a key player in any decarbonization strategy even though it is controversial. To both address this controversy and maintain nuclear power's position in our decarbonization technology stack, we will need to develop and implement new kinds of nuclear reactors that are more sustainable, economically competitive, safe, and reliable than their predecessors.

1.3 Molten Salt Reactors

The first generation of nuclear power reactors includes the first prototypes and early civil deployments in pursuit of cheap and bountiful energy. The second generation of reactors were built on top of this momentum. Following the well publicized and documented accidents at Three Mile Island (1976) and Chernobyl (1981) power plants, the third generation of nuclear power developed with increased safety and reactor lifetime in mind. The fourth generation of nuclear power reactors will need to rapidly respond to increasing electricity consumption and the need for decarbonzation. This means the fourth generation of nuclear reactors need to be built quickly, have widespread use, all while maintaining or increasing fuel efficiency, economic competitiveness, safety, and proliferation resistance.

This is essentially the conclusion that the Generation IV Interational Forum (GIF) – a "co-operative international endeavor seeking to develop the feasibility and performance of fourth generation nuclear systems" [5] – came to in their 2002 roadmap [15]. This effort selected six reactor technologies in total. One of these technologies, the MSR[6], is of particular interest due to the unique challenges and opportunities is presents.

The MSR is so named as it is a nuclear reactor that uses a mixture of liquid-phase salts as a coolant. The use of a liquid fuel enables adoption of *on-line reprocessing* by pumping used fuel out of the reactor and pumping fresh or reprocessed fuel back into the reactor[7]. This enables removal of undesirable fission products – neutron absorbers in particular – produced by the fission reaction. In contrast, while solid-fueled reactors will shuffle the fuel within the assembly over time, the fission products remain trapped in the fuel which can reduce the thermal performance of the fuel via thermal cracking as well as absorb neutrons that would otherwise contribute to the fission reaction.

[5] This is a generalization as this issue is incredibly complex and requires more space that I can give to it in this thesis to fully appreciate

[6] for this thesis *Molten Salt Reactor* refers specifically to the reactor type where the fissile material is dissolved in in the salt coolant

[7] this concept is known as *separations and feeds*

MSRs face several technical and logistical challenges before they can deploy for civilian power generation. High temperature liquid-phase salt will steadily corrode metals over time, so the reactor vessel of a MSR must use special corrosion-resistant materials. High temperature salt itself is extremely hazardous and reacts explosively with moisture, so special procedures and PPE must be used when handling fuel salt. Remote handling of fuel salt may be necessary in some or all instances. Even with these challenges, I believe the potential benefits of MSR technology merit its development.

Modelling and simulation (M&S) codes will play a critical role in licensing GenIV reactors. In preparation for MSRs, both the Department of Energy Office of Nuclear Energy (DOE-NE) and the Nuclear Regulatory Commission (NRC) have identified several technical gaps in current M&S tools that are necessary for efficient and effective license application reviews [9] [38]. In particular, both the DOE-NE and the NRC have identified the fuel composition and its evolution in a MSR to be a key software feature necessary for accident analysis.

1.4 MSR Depletion Codes

To model the changing fuel composition in a MSR, there are at least two processes we must consider:

1. Fuel *depletion*[8]

2. Removal and feed processes

The Bateman equation describes the rate of change of the number density of a nuclide in a nuclear reaction mathematically:

[8]the consumption of fissile material in the fuel and production of fission products via the fission chain reaction

$$\frac{dN_i}{dt} = \sum_j l_{j\to i}\lambda_j N_j + \gamma_i \Sigma^f \phi + \phi N_{i-1}\sigma^a_{i-1} - \lambda_i N_i - N_i \sigma a_i \phi \tag{1.1}$$

$$\begin{aligned}
N_* &= \text{number density for nuclide } * \; [cm^{-3}] \\
l_{j\to i} &= \text{branching ratio for decay mode of nuclide } j \text{ that produces nuclide } i \\
\lambda &= \text{decay constant of nuclide } \; [s^{-1}] \\
\gamma_i &= \text{fission yield fraction for nuclide } i \\
\Sigma^f &= \text{average macroscopic fission cross section } [cm^{-1}] \\
\phi &= \text{neutron flux } [cm^{-2}s^{-1}] \\
\sigma^a_{i-1} &= \text{neutron absorption cross-section for nuclide } i-1 \; [cm^{-2}]
\end{aligned} \tag{1.2}$$

This equation sufficiently describes the process of depletion for any nuclide i. When solving for n nuclides, we solve the matrix problem $\frac{d}{dt}N = \mathbf{A}N$, where N is a n-vector and $\mathbf{A}$ contains all the coefficient terms in equation 1.1. Incorporating removals and feeds into this equation involves the addition of a time dependent removal factor $r_i(t)$ and feed factor $f_i(t)$ to equation 1.1. The resulting equation describes *continuous reprocessing*. For n nuclides, the matrix problem is then $\frac{d}{dt}N = \mathbf{A}N + S(t)$, where S is an n-vector containing the sums of the removal and feed terms for each nuclide i. The additional term in the Bateman equation from continuous reprocessing increases computational cost and implementation difficulty, and may require a different set of preconditioners for numerical stability and convergence.

Alternatively, one could run a depletion simulation, perform the removals and feeds in an external application on the resulting material composition,and run another depletion simulation on the reprocessed composition. This procedure models *batch-wise reprocessing*, in which material from the core is reprocessed at regular intervals rather than continuously. This requires coupling to an external piece of software, which comes with its own challenges, but benefit of this approach is that the external software could support *any* software capable of doing depletion calculations.

SaltProc [35], the focus of this thesis, uses a batch-wise reprocessing approach to model the fuel composition in an MSR and uses Serpent2 [19] to run depletion simulations. SaltProc is unique among its peers as an open source project[9].

[9]There are other software projects that model fuel composition in MSRs. Notably, ChemTriton [10] – a python script for SCALE/TRITON– is functionally similar to SaltProc. Section 1.2 in [34] and section 4.2 in [33] provide a high-level summary of other recent efforts

Historically, software used in licensing, research and development (R&D), and education and training (E&T) efforts in the nuclear field has been closed source and proprietary. For R&D and E&T efforts in particular, this can bring collaborative efforts to a grinding halt until regulatory bodies grant software licenses. Using closed code (CC)s in scientific publications and research presents barriers to reproducibility and the ability of external verification of results.

Regulatory bodies will require new software features (and in some cases entirely new software tools) in order to effectively and efficiently perform licensing activities for the next generation of advanced reactor designs [38], and many of the open source tools emerging in the past decade (e.g. OpenMC [31]) have the advantage over their legacy CC ancestors (e.g. Serpent2 [19]) of using best-practices for software development. It follows that these features and tools are more readily implementable in these new open source projects than in the legacy closed codes.

We are entering the era of Open source software (OSS) purpose-built for applications in nuclear science and engineering. The number of open source projects in this industry (ONIX [13], OpenMC, NJOY21 [6], Cyclus [4], to name a few[10]) is growing in recognition of the need for distributable, high-quality, and transparent software tools. This is perhaps best seen in the International Atomic Energy Agency (IAEA) facilitated Open-source Nuclear COdes for REactor analysis (ONCORE) initiative [16] to "[promote] development and application of open- source multi-physics simulation tools to support research, education, and training for analysis of advanced nuclear power reactors" [18].

1.5 Objectives

While SaltProc itself is open source, Serpent2 is not. OpenMC recently added a `deplete` module for fuel depletion simulations, meaning it is now possible to have a fully open-source stack of dependencies for SaltProc. It is in this spirit that I have added support for OpenMC to SaltProc. This improves the accessibility and usability of SaltProc, and I hope that researchers in this field will begin using and contributing to the tool.

This Master's thesis has two primary objectives

1. Refactor SaltProc for to enable OpenMC support, as well as enable easier implementation for other monte-carlo depletion codes in the future.

2. Verify the implementation on a well studied MSR system.

[10]the awesome-nuclear repository on GitHub [29] has good list of nuclear-related open source software projects.

It is my hope that with this advancement, SaltProc can be both a general purpose research tool as well as assist in answering questions regarding fuel composition evolution in an MSR for licensing purposes.

The structure of this thesis is as follows:

- Chapter 2 describes the structure of SaltProc and process of implementing support for OpenMC
- Chapter 3 describes the reactor design for verification purposes and specifies the inputs and settings.
- Chapter 4 presents the results of the verification study and discusses their implications.
- Chapter 5 summarizes the results, their implications, and suggests avenues for future work.

Chapter 2

Molten Salt Reactor Modeling

Much of our knowledge about MSRs come from experiements on a test reactor called the Molten Salt Reactor Experiment (MSRE) conducted at Oak Ridge National Laboratory (ORNL) in the 1960s, which demonstrated the viability of the MSR concept for use in civillian power programs [17] [32].

The MSRE reactor to this day remains one of the few MSRs to operate.

As of writing this thesis, there are no MSRs currently in operation; we must rely on computational and/or surrogate models to further study MSR physics. The MSRE is a popular choice for computational models due to the availability of experimental data to compare results against. For example, Roelofs et al used the system thermal hydraulics code SPECTRA to model steady state parameters of the fuel-salt, Delayed neutron precursor (DNP) drift, and various fission product behaviors in comparison with actual MSRE data [28]. Podila et al performed a Computational fluid dynamics (CFD) simulation of the MSRE core to investigate the ability of CFD to predict 3D effects in this kind of reactor[26]. In addtion to the MSRE, the Molten Sodium Fast Reactor (MSFR)[21] and Molten Salt Breeder Reactor (MSBR)[27] conceptual reactors are well developed and are actively used in research. Park et al performed a whole core analysis of the MSBR using MCNP6 with additional depletion and reprocessing using CINDER90 and a custom Python script [25]. Aufiero et al extended Serpent2 with online fuel reprocessing capabilities to study depletion in the MSFR [7].

These efforts illustrate that MSR modeling encompasses a wide range of physics domains.

2.1 Modeling depletion in MSRs

Recall in Sections 1.3 and 1.4 we introduced the concepts of fuel depletion and removal and feed processes, and established the importance of modeling fuel depletion to MSR licensing. Depletion codes in the past have shown good behavior when compared with depletion measurements from commercial and research reactors; the TRITON module in the SCALE reactor physics suite couples SCALE's neutron transport solver (Shift) and bateman equation solver (ORIGEN) to get accurate depletion for various reactor core geometries [14];

The Serpent2 monte carlo particle transport code also has fuel depletion capabilities [20]. More recently, OpenMC version 0.11 added support for fuel depletion and shows good agreement with Serpent 2[30].

Depletion simulations on MSRs are difficult to verify through direct measurement due to the lack of a physically operating reactor. Even so, simulations will give us a good idea of what kinds of nuclides we can expect to show up. Table 2.1 summarizes some currently available software tools that can model depletion in MSRs[1].

Table 2.1: Software tools that can model MSR depletion with fuel reprocessing

Software tool	Description	Reprocessing scheme	Citation
ADDER	Interface code with internal depletion capabilities	Component-based continuous	[23]
SCALE	Suite code	Continuous	[8]
SaltProc	Interface code	Component-based batch-wise	[35]

SaltProc has been used to perform lifetime depletion analysis on full-core models of the MSBR[36] and the Transatomic Power MSR[34]. As seen in Figure 2.1, the BOL and EOL neutron flux spectra for the MSBR results qualitatively matches those reported Park et. al [25], as well as with data from the original MSBR report from Robertson et. al [27].

Rykhlevskii compared the TAP MSR results against an ORNL technical report on neutronic performance of the same reactor[11] using identical cross section libraries and model parameters wherever possible. Betzler et. al. used the ChemTriton[10] interface code that perfoms batch-wise reprocessing using the SCALE suite for coupled transport-depletion. ChemTriton was extended to use the Shift[12] monte carlo particle transport tool for 3D neutron transport and depletion [2]. As seen in Figure 2.2, neutron flux spectra results between the SaltProc and ChemTriton simulations showed decent agreement.

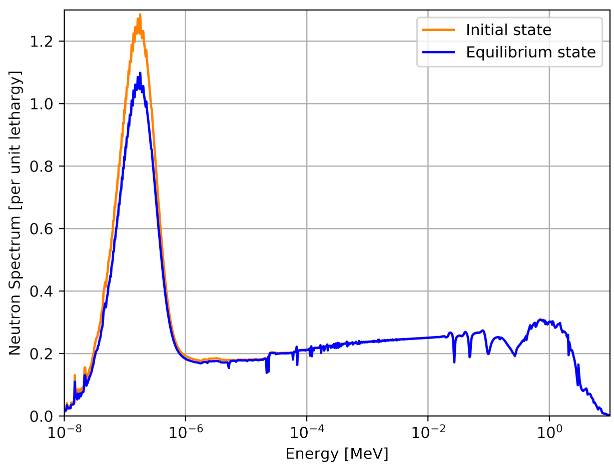
While SaltProc has not been compared to SCALE or ADDER, these results verify that SaltProc produces results that are comparable to those of similar tools.

2.1.1 Practical differences in continuous and batch-wise reprocessing

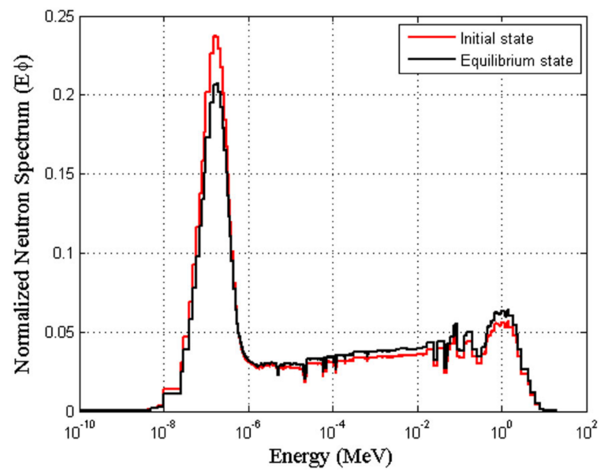
We defined continuous and batch-wise reprocessing and discussed their theoretical differences in in Section 1.4. There are also some practical implications on problem setup and results between these two methods. Betzler, et. al. (2019) [8] perfomed a code-to-code comparsion between new functionality added to the ORIGEN and SCALE/TRITON codes for simulating depletion in MSRs with continuous reprocessing, and the older

[1]additional works like Aufiero et al [7], while notable and relevant, are bespoke modifications to currently existing software, and so cannot truly be considered “available” until the modifications become part of the software

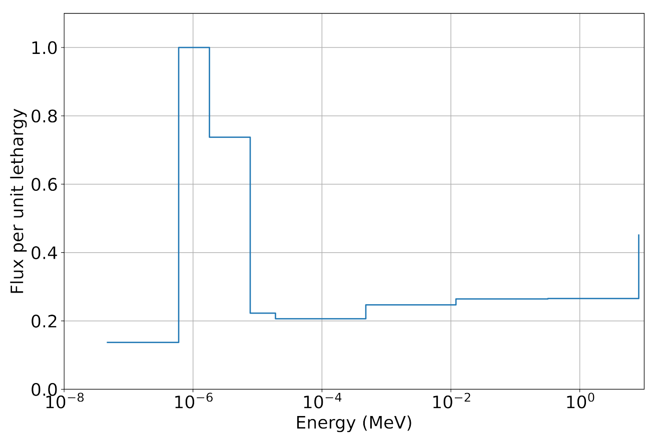
[2]at the time of publication, Shift had not yet been incorporated into SCALE



(a)

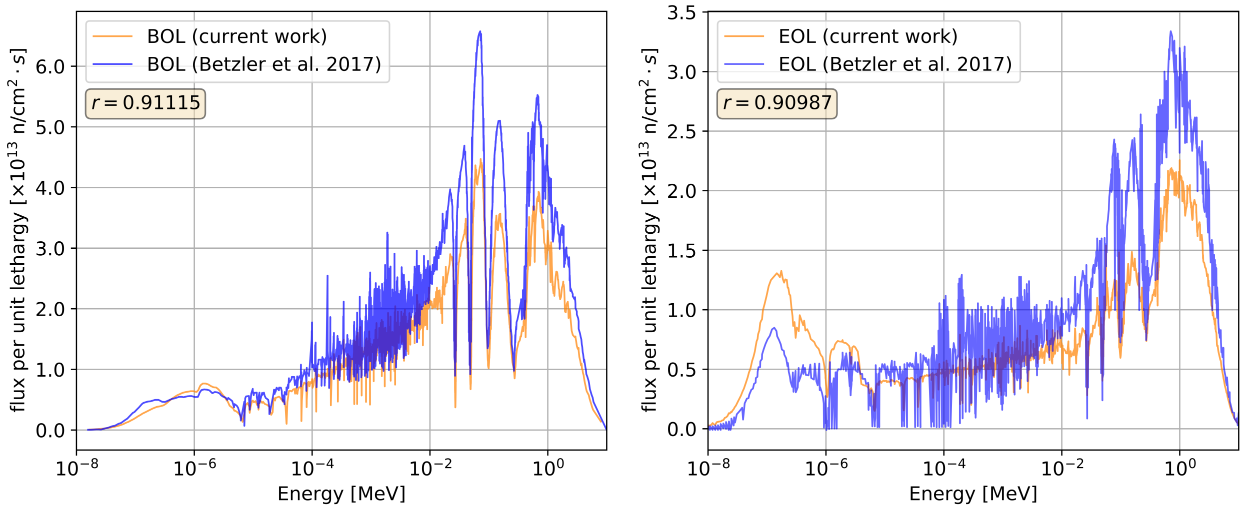


(b)



(c)

Figure 2.1: (a) Neutron flux energy spectrum at initial and equilibrium states normalized by unit lethargy. Reproduced from [36]; (b) Neutron flux spectrum at initial and equilibirum states. Reproduced from [25]; (c) Neutron flux spectrum at the midplane of the MSBR design normalized by the peak flux. Made using data in Table B.2 in [27]



(a) (b)

Figure 2.2: Comparative results of TAP MSR flux spectra at BOL and EOL: 2.2a BOL; 2.2b EOL; Both figures reproduced from [34]

ChemTriton [10] interface code that perfoms batch-wise reprocessing using the SCALE suite for coupled transport-depletion. They found that the two reprocessing schemes generally produced similar results with the following caveats:

1. k_∞ for batch-wise reprocessing trends slightly lower than continuous reprocessing. The authors claim this is due to SCALE's *middlestep depletion method*.

2. The batch-wise method generally results in higher concentrations of nuclides. Notably, nuclides with short half lives, like ^{135}Xe, trends significantly higher in batch-wise reprocessing.

2.2 Differences between Serpent and OpenMC

As mentioned in Section 1.5, a major objective of this thesis is to implement support for OpenMC's `deplete` module. In this context, I would like to discuss some of the differences between OpenMC and Serpent2. The most in-depth discission of these differences that I have found are in [30]. I have tabulated the differences discussed in that paper below. As mentioned earlier, Romano found good agreement between OpenMC depletion results and Serpent depletion results, but notes that one must take careful consideration of the data and internal settings to get good results between codes. Of note are the default energy release per fission of ^{235}U used for power normalization and capture branching ratios. We follow Romano's methodology to ensure data consitency in our comparative study. The key features to consider:

Table 2.2: Differences between OpenMC and Serpent

Metric	OpenMC	Serpent
Access	Open source, hosted on GitHub	Export controlled, code distributed by RSICC (USA), OECD/NEA
User interface	Python API	Manual writing of geometry, material, and settings files
Handling of cross sections in between library temperatures	interpolation between neighboring temperatures	target motion-sampling [39]
Matrix exponential solver	IPF CRAM-48	PDF CRAM-14
Cross section file formate	HDF5	ACE
$S(\alpha, \beta)$ representation	continuous	tabulated
Depletion data source	ENDF decay and fission product yield sublibraries	depletion chain XML file

- Utilize identical cross-section libraries, depletion data, and capture branching ratios
- Use the same fission heating values.

We will discuss our procedure for ensuring data consitency in Chapter 4.

Appendix

Appendix.

References

[1] (2021). 5 Fast Facts About Nuclear Energy. url accessed 2022-02-15: https://www.energy.gov/ne/articles/5-fast-facts-about-nuclear-energy.

[2] (2021a). Electricity generation, capacity, and sales in the United States. url accessed 2022-02-15: https://www.eia.gov/energyexplained/electricity/electricity-in-the-us-generation-capacity-and-sales.php.

[3] (2021b). Frequently Asked Questions (FAQs). url accessed 2022-02-15: https://www.eia.gov/tools/faqs/faq.php.

[4] (2022a). Cyclus. url accessed 2022-02-14: https://github.com/cyclus/cyclus.

[5] (2022). GIF Portal - Portal Site Public Home. url accessed 2022-01-17: https://www.gen-4.org/gif/.

[6] (2022b). NJOY21 — NJOY for the 21st Century. url accessed 2022-02-14: https://github.com/njoy/NJOY21.

[7] Aufiero, M., Cammi, A., Fiorina, C., Leppänen, J., Luzzi, L., and Ricotti, M. (2013). An extended version of the SERPENT-2 code to investigate fuel burn-up and core material evolution of the Molten Salt Fast Reactor. *Journal of Nuclear Materials*, 441(1-3):473–486.

[8] Betzler, B. R., Bekar, K. B., Wieselquist, W., Hart, S. W., and Stimpson, S. G. (2019a). Molten Salt Reactor Fuel Depletion Tools in SCALE. In *Proc. of GLOBAL International Fuel Cycle Conference*, Seattle, WA, United States. American Nuclear Society.

[9] Betzler, B. R., Heidet, F., Feng, B., Rabiti, C., Sofu, T., and Brown, N. R. (2019b). Modeling and simulation functional needs for molten salt reactor licensing. *Nuclear Engineering and Design*, 355:110308.

[10] Betzler, B. R., Powers, J. J., and Worrall, A. (2017a). Molten salt reactor neutronics and fuel cycle modeling and simulation with SCALE. *Annals of Nuclear Energy*, 101:489–503.

[11] Betzler, B. R., Robertson, S., Davidson, E. E., Powers, J. J., Worrall, A., Dewan, L., and Massie, M. (2017b). Assessment of the Neutronic and Fuel Cycle Performance of the Transatomic Power Molten Salt Reactor Design. Technical Report ORNL/TM-2017/475 CRADA/NFE-16-06345, Oak Ridge National Lab.(ORNL), Oak Ridge, TN (United States).

[12] Davidson, G. G., Pandya, T. M., Johnson, S. R., Evans, T. M., Isotalo, A. E., Gentry, C. A., and Wieselquist, W. A. (2018). Nuclide depletion capabilities in the Shift Monte Carlo code. *Annals of Nuclear Energy*, 114:259–276.

[13] de Troullioud de Lanversin, J., Kütt, M., and Glaser, A. (2021). ONIX: An open-source depletion code. *Annals of Nuclear Energy*, 151:107903.

[14] DeHart, M. D. and Bowman, S. M. (2011). Reactor Physics Methods and Analysis Capabilities in SCALE. *Nuclear Technology*, 174(2):196–213.

[15] DOE-NE and GIF (2002). A Technology Roadmap for Generation IV Nuclear Energy Systems. Technical Report GIF-002-00, 859029.

[16] Fiorina, C., Shriwise, P., Dufresne, A., Ragusa, J., Ivanov, K., Valentine, T., Lindley, B., Kelm, S., Shwageraus, E., Monti, S., Batra, C., Pautz, A., Lorenzi, S., Rubiolo, P., Clifford, I., and Dechenaux, B. (2021). AN INITIATIVE FOR THE DEVELOPMENT AND APPLICATION OF OPEN-SOURCE MULTI-PHYSICS SIMULATION IN SUPPORT OF R&D AND E&T IN NUCLEAR SCIENCE AND TECHNOLOGY. *EPJ Web of Conferences*, 247. Publisher: EDP Sciences.

[17] Haubenreich, P. N. and Engel, J. R. (1970). Experience with the Molten-Salt Reactor Experiment. *Nuclear Technology*, 8(2):118–136.

[18] IAEA (2022). Open-source Nuclear Codes for Reactor Analysis - Home. url accessed 2022-01-25: `https://nucleus.iaea.org/sites/oncore/SitePages/Home.aspx`.

[19] Leppanen, J., Pusa, M., Viitanen, T., Valtavirta, V., and Kaltiaisenaho, T. (2014). The Serpent Monte Carlo code: Status, development and applications in 2013. *Annals of Nuclear Energy*, 82:142–150.

[20] Leppänen, J. and Pusa, M. (2009). Burnup calculation capability in the PSG2/Serpent Monte Carlo reactor physics code. *Proc. M&C*, pages 3–7.

[21] Merle-Lucotte, E., Heuer, D., Allibert, M., Brovchenko, M., Capellan, N., and Ghetta, V. (2011). Launching the thorium fuel cycle with the Molten Salt Fast Reactor. In *Proceedings of ICAPP*, pages 2–5.

[22] Mitchell, J. F. B. (1989). The "Greenhouse" effect and climate change. *Reviews of Geophysics*, 27(1):115–139. _eprint: `https://onlinelibrary.wiley.com/doi/pdf/10.1029/RG027i001p00115`.

[23] Nelson, A. G., Chee, G. J. Y., and Jarrett, M. G. (2021). MOLTEN SALT REACTOR DEPLETION TECHNIQUES IN THE ADDER REACTOR DEPLETION AND FUEL MANAGEMENT ANALYSIS CODE. In *Framework Development for Reactor Analysis*, page 10, Raleigh, North Carolina.

[24] Paola A. Arias, Nicolas Bellouin, Erika Coppola, Richard Jones, Gerhard Krinner, Jochem Marotzke, Vaishali Naik, Matthew D. Palmer, Gian-Kasper Plattner, Joeri Rogelj, Maisa Rojas, Jana Sillmann, Trude Storelvmo, Peter W. Thorne, Blair Trewin, Krishna Achuta Rao, Bhupesh Adhikary, Richard P. Allan, Kyle Armour, Govindasamy Bala, Rondrotiana Barimalala, Sophie Berger, Josep G. Canadell, Christophe Cassou, Annalisa Cherchi, William Collins, William D. Collins, Sarah L. Connors, Susanna Corti, Faye Cruz, Frank J. Dentener, Claudine Dereczynski, Alejandro Di Luca, Aida Diongue Niang, Francisco J. Doblas-Reyes, Alessandro Dosio, Hervé Douville, François Engelbrecht, Veronika Eyring, Erich Fischer, Piers Forster, Baylor Fox-Kemper, Jan S. Fuglestvedt, John C. Fyfe, Nathan P. Gillett, Leah Goldfarb, Irina Gorodetskaya, Jose Manuel Gutierrez, Rafiq Hamdi, Ed Hawkins, Helene T. Hewitt, Pandora Hope, Akm Saiful Islam, Christopher Jones, Darrell S. Kaufman, Robert E. Kopp, Yu Kosaka, James Kossin, Svitlana Krakovska, June-Yi Lee, Jian Li, Thorsten Mauritsen, Thomas K. Maycock, Malte Meinshausen, Seung-Ki Min, Pedro M.S. Monteiro, Thanh Ngo-Duc, Friederike Otto, Izidine Pinto, Anna Pirani, Krishnan Raghavan, Roshanka Ranasinghe, Alex C. Ruane, Lucas Ruiz, Jean-Baptiste Sallée, Bjørn H. Samset, Shubha Sathyendranath, Sonia I. Seneviratne, Anna A. Sörensson, Sophie Szopa, Izuru Takayabu, Anne-Marie Treguier, Bart van den Hurk, Robert Vautard, Karina von Schuckmann, Sönke Zaehle, Xuebin Zhang, and Kirsten Zickfeld (2021). 2021: Technical Summary. Technical report, Cambridge University Press.

[25] Park, J., Jeong, Y., Lee, H. C., and Lee, D. (2015). Whole core analysis of molten salt breeder reactor with online fuel reprocessing: Whole core analysis of MSBR with online fuel reprocessing. *International Journal of Energy Research*, pages n/a–n/a.

[26] Podila, K., Chen, Q., and Rao, Y. (2019). CFD Simulations of Molten Salt Reactor Experiment Core. *Nuclear Science and Engineering*, 193(12):1379–1393. Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/00295639.2019.1627177.

[27] Robertson, R. C. (1971). Conceptual Design Study of a Single-Fluid Molten-Salt Breeder Reactor. Technical Report ORNL–4541, ORNL.

[28] Roelofs, F. and Stempniewicz, M. M. (2021). Molten salt modelling capabilities in SPECTRA and application to MSRE and Mk1-PB-FHR. *Nuclear Engineering and Design*, 381:111360.

[29] Romano, P. (2022). Awesome Nuclear. url accessed 2022-02-14: `https://github.com/paulromano/awesome-nuclear`.

[30] Romano, P. K. (2021). Depletion capabilities in the OpenMC Monte Carlo particle transport code. *Annals of Nuclear Energy*, page 15.

[31] Romano, P. K., Horelik, N. E., Herman, B. R., Nelson, A. G., Forget, B., and Smith, K. (2015). OpenMC: A state-of-the-art Monte Carlo code for research and development. *Annals of Nuclear Energy*, 82(Supplement C):90–97.

[32] Rosenthal, M. W., Kasten, P. R., and Briggs, R. B. (1970). Molten-Salt Reactors - History, Status, and Potential. *Nuclear Applications and Technology*, 8(2):107–117.

[33] Rykhlevskii, A. (2018). Advanced online fuel reprocessing simulation for Thorium-fueled Molten Salt Breeder Reactor. Master's thesis, University of Illinois at Urbana-Champaign, Urbana, IL.

[34] Rykhlevskii, A. (2020). *Fuel Processing Simulation Tool for Liquid-fueled Nuclear Reactors*. Doctoral Dissertation, University of Illinois at Urbana-Champaign, Urbana, IL.

[35] Rykhlevskii, A., Bae, J. W., and Huff, K. (2018). arfc/saltproc: Code for online reprocessing simulation of molten salt reactor with external depletion solver SERPENT. *Zenodo*.

[36] Rykhlevskii, A., Bae, J. W., and Huff, K. D. (2019). Modeling and simulation of online reprocessing in the thorium-fueled molten salt breeder reactor. *Annals of Nuclear Energy*, 128:366–379.

[37] Szilard, R., Sharpe, P., Kee, E., Davis, E., and Grecheck, E. (2016). Economic and Market Challenges Facing the U.S. Nuclear Commercial Fleet. Technical Report INL/EXT-16-39951, Idaho National Lab. (INL), Idaho Falls, ID (United States).

[38] U.S. Nuclear Regulatory Comission (2020). NRC Non-Light Water Reactor Vision and Strategy, Volume 1 - Computer Code Suite for Non-LWR Plant Systems Analysis. Technical Report ML20030A176.

[39] Viitanen, T. and Leppänen, J. (2012). Explicit Treatment of Thermal Motion in Continuous-Energy Monte Carlo Tracking Routines. *Nuclear Science and Engineering*, 171(2):165–173. Publisher: Taylor & Francis _eprint: https://doi.org/10.13182/NSE11-36.