Chapter 6

Future Work and Proposed Simulations

The need for this work is shown by a summary of how additive manufacturing of nuclear reactor core components frees complex reactor geometries from previous manufacturing constraints enabling reactor designers to reexamine reactor core design optimization. The literature review (Chapter 2) concluded that stochastic evolutionary algorithm optimization methods could find global optimums for reactor design problems in the vast exploration design space enabled by additive manufacturing. Chapter 3 introduced the Fluoride-Salt-Cooled High-Temperature Reactor (FHR) benchmark (AHTR design) and highlighted the reactor's benefits, such as passive safety behavior with negative temperature coefficients. Chapter 4 introduced the Reactor Evolutionary Algorithm Optimizer (REALM) software package which applies evolutionary algorithm optimization techniques to nuclear reactor design. Chapter 5 demonstrated successfully applying REALM to optimize the TRISO packing fraction distribution in an AHTR slab.

Based on the preliminary work conducted, this chapter proposes future simulations categorized into two groups: AHTR development and REALM optimization. The proposed work aims to address AHTR modeling challenges further and demonstrate using REALM for multi-objective AHTR optimization of arbitrary geometries and fuel distribution. For AHTR development, I propose the following simulations:

- AHTR 3D full core neutronics OpenMC simulation
- AHTR fuel slab and one-third fuel assembly multiphysics Moltres simulation

For REALM optimization, I propose the following REALM simulations:

- AHTR slab geometry optimization to maximize k_{eff}, minimize power peaking, and maximize heat transfer by varying TRISO x-axis distribution and FLiBe channel shape using OpenMC.

- AHTR one-third fuel assembly optimization to maximize k_{eff}, minimize power peaking, and maximize heat transfer by varying TRISO XY axes distribution and FLiBe channel shape using OpenMC.

6.1 AHTR Model Development

The FHR benchmark introduced in Chapter 3 is an ongoing NEA project to assess the modeling and simulation capabilities for the AHTR. Benchmark participants', including the UIUC team, contributed Phases I-A and I-B (2D assembly steady-state and depletion) so far. The upcoming phases consist of 3D neutronics models and multiphysics models. Thus, to support the FHR benchmark, the proposed work will complete the benchmark's Phase I-C. In preparation for the later multiphysics benchmark phases, the proposed work will utilize Moltres to model AHTR multiphysics.

6.1.1 FHR Benchmark Phase I-C

The FHR benchmark's Phase I-C extends the 2D assembly model from Phases I-A and I-B into a 3D assembly model. The benchmark organizers will release Phase I-C's detailed specifications and required results in June 2021.

6.1.2 AHTR Multiphysics Model

Setting up a Moltres multiphysics simulation requires the user to provide group constant data from a neutron transport solver, such as OpenMC. The group constants used for neutronics calculations in Moltres are [49, 58]:

Σ_g^f: macroscopic fission cross section in group g,

Σ_g^r: macroscopic removal cross section in group g,

$\Sigma_{g'\to g}^s$: macroscopic scattering cross section from group g' to g,

D_g: diffusion coefficient of neutrons in group g,

ϵ_g: average fission energy per fission by a neutron from group g,

ν: average neutron yield per fission by a neutron from group g,

$\frac{1}{v}$: inverse neutron speed in group g,

λ_i: decay constant of delayed neutron precursor (DNP) group i,

β_{eff}: effective delayed neutron fraction.

These group constants are extracted from the neutron transport solver's output files using a Python script from the Moltres Github repository [48]. The Python script currently enables group constant extraction from Serpent [45] and SCALE [10] output files. I used OpenMC to model the neutronics of the AHTR for the FHR benchmark; thus, I will add the capability to extract group constants from OpenMC output files to the Moltres Python group constants extraction script.

Section 5.2 demonstrated that multigroup neutronics simulation with four-group energy and spatial homogenization of the AHTR fuel slab generated a k_{eff} within uncertainty of the continuous energy and space neutronics simulation. I will utilize these homogenizations to create group constants for the Moltres AHTR fuel slab simulation. I will then set up a mesh for the AHTR fuel slab and run a Moltres simulation and verify Moltres' ability to reproduce the following key neutronics parameters:

- k_{eff} (effective multiplication factor)

- reactivity coefficients: β_{eff}, α_D (doppler coefficient), $\alpha_{T,FliBe}$ (FLiBe temperature co-

Table 6.1: Reactor Evolutionary Algorithm Optimizer (REALM) optimization problem objectives with their quantification descriptions.

Objective	Quantification
Best neutronics	Maximize k_{eff}
Maximize heat transfer	Maximize ϕ_{total} in areas along FLiBe coolant
Minimize power peaking	Minimize $P_{high} - P_{low}$

efficient), α_M (moderator temperature coefficient)

- Neutron energy spectrum
- $\phi_1(\vec{x},\vec{y}), \phi_2(\vec{x},\vec{y}), \phi_3(\vec{x},\vec{y})$ (neutron flux distribution in four coarse energy groups)

Once verified, I will run a steady-state Moltres multiphysics simulation to determine the maximum temperature in the fuel slab at steady-state.

With information gleaned from the Moltres AHTR fuel slab simulation, I will test out energy and spatial homogenization for generating group constants for a one-third AHTR fuel assembly model. Then, proceed to set up the one-third AHTR fuel assembly model simulation, verify its key neutronics parameters, and finally run a steady-state Moltres simulation.

6.2 REALM Optimization

Section 5.1 concluded that the AHTR slab optimization problem should be further developed by considering other objectives such as maximizing heat transfer and minimizing power peaking in the core. In the proposed work, I will explore each objective separately and then together. Table 6.1 describes each objective and how the objective will be quantified. The slab parameters that will be varied to meet the described problem objectives include:

- TRISO particle packing fraction distribution
- FLiBe coolant channel shape

Table 6.2: Proposed Reactor Evolutionary Algorithm Optimizer (REALM) simulations for optimizing Advanced High Temperature Reactor (AHTR) fuel assembly. Simulations explore two geometries: straightened AHTR fuel slab and AHTR's diamond-shaped section containing six fuel slabs.

Simulation	AHTR Geometry	Objectives	Varying Parameters
1	Single fuel slab	• Maximize k_{eff}	• TRISO distribution
2	Single fuel slab	• Maximize heat transfer	• TRISO distribution
3	Single fuel slab	• Minimize power peaking	• TRISO distribution
4	Single fuel slab	• Maximize k_{eff}	• FLiBe channel shape
5	Single fuel slab	• Maximize heat transfer	• FLiBe channel shape
6	Single fuel slab	• Minimize power peaking	• FLiBe channel shape
7	Single fuel slab	• Maximize k_{eff} • Maximize heat transfer • Minimize power peaking	• TRISO distribution
8	Single fuel slab	• Maximize k_{eff} • Maximize heat transfer • Minimize power peaking	• FLiBe channel shape
9	Single fuel slab	• Maximize k_{eff} • Maximize heat transfer • Minimize power peaking	• TRISO distribution • FLiBe channel shape
10	Diamond section with six fuel slabs	• Maximize k_{eff} • Maximize heat transfer • Minimize power peaking	• TRISO distribution
11	Diamond section with six fuel slabs	• Maximize k_{eff} • Maximize heat transfer • Minimize power peaking	• FLiBe channel shape
12	Diamond section with six fuel slabs	• Maximize k_{eff} • Maximize heat transfer • Minimize power peaking	• TRISO distribution • FLiBe channel shape

I will conduct these optimizations for the straightened AHTR fuel slab geometry (as seen in Figure 5.1) and for one diamond-shaped sector containing six fuel slabs (as seen in Figure 3.2) with x-y periodic and z reflective boundary conditions. Table 6.2 outlines the details of the proposed simulations. I will use the optimal hyperparameters derived in Section 5.1.2 for the proposed simulations. Ideally, a new hyperparameter search should be conducted for each simulation to find the best hyperparameter set for each unique problem; however, the computational expense for conducting 11 hyperparameter searches is impractical. Using the same hyperparameter set is acceptable because the problems are similar. Table 6.3

Table 6.3: Hyperparameter values for the best hyperparameter set calculated in Section 5.1.2.

Hyperparameters	**Values**
Population size	60
Generations	10
Mutation probability	0.23
Mating probability	0.46
Selection operator	`selTournament`
Selection individuals	15
Selection tournament size	5
Mutation operator	`mutPolynomialBounded`
Mating operator	cxBlend

summarizes the optimal hyperparameters.

The REALM simulations proposed in Table 6.2 could be extended to include Moltres evaluations if the proposed AHTR multiphysics Moltres simulations (Section 6.1.2) find approximations and assumptions that maintain accuracy while keeping acceptable Moltres runtimes.

6.3 Conclusion

Additive manufacturing of nuclear reactor components is a quickly developing field thanks to breakthroughs in additive manufacturing fabrication of metal components. The promise of cheaper and faster manufacturing of reactor components with additive manufacturing frees complex reactor geometries from previous manufacturing constraints and allows reactor designers to reexamine reactor design optimization. Therefore, I propose to explore the vast design space enabled by additive manufacturing, with the evolutionary algorithm optimization technique that works well to find global optimums in multi-objective design problems such as nuclear reactor optimization.

In the preliminary work, I designed the REALM Python package that applies evolutionary algorithm optimization techniques to nuclear reactor design using the DEAP Python module, OpenMC, and Moltres. The motivation for REALM is to enable reactor designers to utilize robust evolutionary algorithm optimization methods without going through the cumbersome process of setting up a genetic algorithm framework. With the many benefits of AHTRs, I chose to apply the evolutionary algorithm optimization methods to this reactor type. I participated in Phase I-A and I-B of the OECD NEA's FHR benchmarking exercise. I also applied REALM to a single objective function problem: maximize k_{eff} in the AHTR fuel slab by varying the TRISO particle packing fraction distribution. This problem demonstrated the effectiveness and robustness of genetic algorithms at optimizing reactor parameters for an objective function. However, many other objectives should be considered, such as maximizing heat transfer and minimizing power peaking in the core.

Therefore, I propose to further explore using REALM for multi-objective AHTR optimization of arbitrary geometries and fuel distribution. Optimization objectives include maximizing k_{eff}, maximizing heat transfer, and maximizing power peaking. I also propose to further address AHTR modeling challenges by completing Phase I-C of the FHR benchmark and set up Moltres simulations to model AHTR multiphysics.