Computational Modeling of Molten Salt Infiltration and Oxidation in Nuclear Graphite

INL/RPT-25-83943

Advanced Reactor Technologies

MARCH 2025

Veerappan Prithivirajan

Idaho National Laboratory

Idaho National Laboratory

Battelle Energy Alliance manages INL for the U.S. Department of Energy's Office of Nuclear Energy

INL/RPT-25-83943

Computational Modeling of Molten Salt Infiltration and Oxidation in Nuclear Graphite

Veerappan Prithivirajan
Idaho National Laboratory

March 2025

Idaho National Laboratory
Computational Mechanics and Materials Department
Idaho Falls, Idaho 83415

http://www.inl.gov

Prepared for the
U.S. Department of Energy
Office of Nuclear Energy
Under DOE Idaho Operations Office
Contract DE-AC07-05ID14517

Page intentionally left blank

INL ART Program

Computational Modeling of Molten Salt Infiltration and Oxidation in Nuclear Graphite

INL/RPT-25-83943

March 2025

Technical Reviewer: (Confirmation of mathematical accuracy, and correctness of data and appropriateness of assumptions.)

March 18, 2025

Benjamin W. Spencer
Computational Scientist

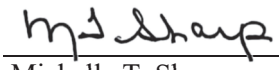
Date

Approved by:

03/19/2025

Kip L. Kleimenhagen
ART Project Manager

Date



3/19/2025

Michelle T. Sharp
INL Quality Assurance

Date

ABSTRACT

Graphite is utilized as a moderator and a reflector in advanced nuclear reactor designs due to its high thermal conductivity, neutron moderation properties, and resistance to radiation damage. However, its long-term performance and reliability are challenged by degradation mechanisms, such as molten salt infiltration in molten salt reactors (MSRs) and oxidation in gas-cooled reactors (GCRs). These mechanisms can compromise the structural integrity and operational lifetime of graphite components, necessitating a more detailed assessment of their physical behavior. This report focuses on the development of computational models for molten salt infiltration and graphite oxidation to aid in the design and performance analysis of graphite components.

For molten salt infiltration, a computational framework is developed that couples incompressible Navier-Stokes and phase-field models to simulate the penetration of molten salt into the interconnected pore structure of graphite. Initial model verification is performed using two-phase flows in two dimensions, thus demonstrating the ability of the model to capture fundamental physical behavior and agree with the analytical solution. This framework is then applied to a realistic IG110 nuclear graphite, where a computed tomography extracted pore geometry is used to analyze the infiltration behavior of FLiNaK molten salt. This model provides insights into how the microstructure and other relevant parameters influence the transport pathways of molten salt into graphite, potentially offering a means to rapidly evaluate a graphite grade's resistance to infiltration.

For oxidation, the report details pore-scale mass and heat transport models, describing the diffusion of gases, reaction kinetics, and thermal effects. Additionally, this report highlights the inconsistencies in the existing volume-averaged macroscopic model, particularly in the upscaling of reaction kinetics and flux terms, and surface to volume transformations. These inconsistencies suggest that current formulations may not accurately capture the experimentally observed graphite oxidation process, highlighting the need for improved model development.

This work advances the development of physics-based computational models for graphite degradation, contributing to improved predictive models for next-generation nuclear reactor designs. Future efforts will focus on refining the infiltration model to address non-physical behaviors and enhance its robustness. Additionally, for oxidation, further studies will employ the principles of volume averaging to rigorously derive the upscaled equations, potentially in collaboration with subject matter experts.

ACKNOWLEDGMENTS

The author would like to recognize Dr. José D Arregui Mena from Oak Ridge National Laboratory (ORNL) for providing a computed tomography slice of IG110 nuclear graphite used for the simulations in this report. The author also thanks Dr. Nidia Gallego from ORNL for the insightful discussions she offered on molten salt infiltration work. Additionally, the author appreciates Dr. Mauricio Tano for his assistance with the formulation and Dr. Daniel Schwen for his help with the code implementation, both from Idaho National Laboratory (INL). Furthermore, the author expresses gratitude to his INL colleagues for their active participation in discussions, thoughtful questions, and valuable feedback—all of which significantly contributed to this report: Drs. William Windes, Benjamin Spencer, and Joshua Kane.

This report was authored by a contractor of the U.S. Government under Contract DE-AC07-05ID14517. Accordingly, the U.S. Government retains a non-exclusive, royalty-free license to publish or reproduce the published form of this contribution, or allow others to do so, for U.S. Government purposes. Funding was provided by the Advanced Reactor Technologies (ART) program.

This research made use of the resources of the High Performance Computing Center at INL, which is supported by the Office of Nuclear Energy of the U.S. Department of Energy and the Nuclear Science User Facilities under Contract No. DE-AC07-05ID14517.

Page intentionally left blank

CONTENTS

FIGURES

TABLES

Page intentionally left blank

ACRONYMS

2D	two-dimensional
ART	Advanced Reactor Technologies
BC	boundary condition
CO	carbon monoxide
CO_2	carbon dioxide
CT	computed tomography
GCR	gas-cooled reactor
INL	Idaho National Laboratory
MOOSE	Multiphysics Object Oriented Simulation Environment
MSR	molten salt reactor
MSRE	Molten Salt Reactor Experiment
NRC	U.S. Nuclear Regulatory Commission
O_2	oxygen
ORNL	Oak Ridge National Laboratory
YL	Young-Laplace

Page intentionally left blank

1. INTRODUCTION

Graphite is being considered as a moderator and a reflector in several reactor designs, due to its excellent thermal conductivity, high-temperature stability, good resistance to radiation damage, and neutron moderation capabilities. However, the long-term performance and reliability of graphite components are challenged by various degradation mechanisms that depend on specific reactor environments and operating conditions. Understanding and predicting these degradation processes is essential for ensuring reactor safety, structural integrity, and extended operational lifetimes. This report focuses on two degradation mechanisms affecting graphite in distinct reactor designs: molten salt infiltration in molten salt reactors (MSRs) [1] and oxidation in gas-cooled reactors (GCRs) [2–4].

1.1. Molten Salt Infiltration

In advanced reactor designs such as in MSRs—particularly in fuel-salt-based designs like the Molten Salt Reactor Experiment (MSRE)—graphite components can come directly into contact with the molten salt-fuel [5]. Due to the porous nature of the graphite microstructure, which is unique to each graphite grade, salt can infiltrate into the graphite, potentially causing physical degradation. Collaborative efforts with the U.S. Nuclear Regulatory Commission (NRC) are currently focused on developing computational tools to assess the structural integrity of graphite with salt infiltration. These efforts are ongoing concurrently with this work. However, the critical need still remains for a mechanistic model to predict the extent and progression of salt infiltration for various grades of graphite and operational conditions. Addressing this gap is the primary focus of this report.

1.2. Oxidation

GCRs, another type of advanced reactor, face a different challenge: the oxidation of graphite components. Oxidation can occur during accident scenarios (resulting in acute oxidation) or through chronic exposure to oxygen (O_2) impurities in the reactor environment, originating from trace amounts of air or water vapor. This degradation mechanism leads to material loss and compromises the structural integrity of graphite. Key factors of interest include the rate of mass loss over time and the evolution of bulk density, both of which determine how rapidly graphite degrades under varying temperature and environmental conditions.

An oxidation model was developed and implemented in the Grizzly software to simulate this degradation; however, it exhibits inconsistencies that undermine its predictive capabilities [6]. These issues have prompted a re-evaluation of the model's fundamental formulation to improve its accuracy in predicting oxidation behavior. A critical review of the existing model is presented first, highlighting the potential inconsistencies. Following this review, a pathway to derive more accurate governing equations is presented, which constitutes the second focus of this report.

1.3. Report Organization

The report is organized as follows. Section 2 covers the model development for molten salt infiltration into graphite including the phenomena of infiltration, governing equations, finite element formulation and implementation, basic verification, application, and future work. Section 3 focuses on the physics of oxidation, a pore-scale transport model, a review of the existing model, and future work strategies. Section 4 summarizes the results from this study.

2. MODEL DEVELOPMENT FOR MOLTEN SALT INFILTRATION INTO GRAPHITE

2.1. Phenomena of Molten Salt Infiltration

Molten salt infiltration in graphite occurs through the physical penetration of molten salt into the interconnected pore structure of the graphite. Most molten salts are non-wetting type fluids that make a contact angle greater than 90° with graphite. These salts require a net pressure differential between the open pores at the surface and the bulk to permeate into the pores of the graphite microstructure. This process is also influenced by the size and connectivity of the pores, the surface characteristics of the graphite pore walls, and the physical properties of the molten salts, such as viscosity, surface tension, and temperature. To assess whether infiltration will occur, the Young-Laplace (YL) equation is often used, though this equation does not quantify the extent of the penetration. The YL equation is expressed as:

$$\Delta P = -\frac{4\gamma \cos\theta}{d} \tag{1}$$

where ΔP is the minimum pressure differential required for infiltration, γ is the surface tension of the molten salt, θ is the contact angle between the molten salt and the graphite surface, and d is the pore diameter. A more detailed explanation of this phenomena can be found in an earlier report [1].

While the YL equation determines the conditions which are necessary for molten salt to initiate infiltration, it does not capture the progression or depth of infiltration. A more comprehensive understanding requires advanced modeling approaches that account for fluid dynamics within the pore network, including the effects of viscosity, pore connectivity, and surface tension, etc., to accurately predict the extent of salt penetration into the graphite.

2.2. Governing Equations

The infiltration of molten salt into porous graphite involves the displacement of the cover gas by molten salt within the graphite's pore network. This behavior is modeled using the coupled Navier-Stokes equation for fluid dynamics and the phase-field method for interface tracking between the two phases [7, 8]. The following sections will provide a detailed explanation of the governing equations and their components.

2.2.1. Phase-Field Method

The phase-field method is used to track the interface between the molten salt and the cover gas. This method introduces a phase variable ϕ that smoothly transitions between the two phases:

$$\phi = \begin{cases} 1 & \text{in the molten salt phase,} \\ -1 & \text{in the cover gas phase,} \\ -1 < \phi < 1 & \text{at the interface.} \end{cases}$$

The evolution of ϕ is governed by the Cahn-Hilliard equation [9]:

$$\frac{\partial \phi}{\partial t} + \mathbf{u} \cdot \nabla \phi = \gamma \nabla^2 G \tag{2}$$

where:

- $\mathbf{u}$ is the fluid velocity,
- γ is the mobility coefficient,
- G is the chemical potential.

The chemical potential G is expressed as:

$$G = -\lambda \nabla^2 \phi + \frac{\lambda}{\epsilon^2} \phi(\phi^2 - 1) \tag{3}$$

where:

- λ is the mixing energy density,
- ϵ is a parameter controlling the thickness of the diffuse interface.

The width of the diffuse-interface is given by $W = 2\sqrt{2}\epsilon$. The surface tension σ is given by:

$$\sigma = \frac{2\sqrt{2}}{3} \frac{\lambda}{\epsilon} \tag{4}$$

Substituting the expression for G into the Cahn-Hilliard equation [9] introduces a biharmonic term ($\nabla^4 \phi$), which is computationally challenging to solve directly. To address this, the equation is decomposed into two coupled second-order equations using an auxiliary variable of ψ [7, 8]:

$$\psi = -\epsilon^2 \nabla^2 \phi + (\phi^2 - 1)\phi \tag{5}$$

$$\frac{\partial \phi}{\partial t} + \mathbf{u} \cdot \nabla \phi = \frac{\gamma \lambda}{\epsilon^2} \nabla^2 \psi \tag{6}$$

2.2.2. Fluid Flow Equations

To simulate the flow behavior of the two immiscible fluids (molten salt and cover gas), the phase-field equations are coupled with the incompressible Navier-Stokes and continuity equations. The following represents the incompressible Navier-Stokes equation:

$$\rho(\phi) \left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} \right) = -\nabla p + \nabla \cdot \tau + \mathbf{F}_{\text{st}} + \rho(\phi) \mathbf{g} \tag{7}$$

where:

- ρ is the fluid density,
- p is the pressure,
- τ is the shear stress,
- $\mathbf{F}_{\text{st}}$ is the surface tension force

- $\mathbf{g}$ is the acceleration due to gravity

The surface tension force term $\mathbf{F}_{\text{st}}$ is defined as:

$$\mathbf{F}_{\text{st}} = G\nabla\phi \tag{8}$$

Here, ϕ is the phase-field variable and G is the chemical potential. The surface tension term $\mathbf{F}_{\text{st}}$ serves as a coupling term between the two systems of equations and is obtained as a function of the phase-field variable ϕ.

τ is the shear stress, given by:

$$\tau = \mu(\phi)(\nabla\mathbf{u} + \nabla\mathbf{u}^T) \tag{9}$$

To ensure incompressibility, the continuity equation is applied:

$$\nabla \cdot \mathbf{u} = 0 \tag{10}$$

The material properties of the two-phase system are smoothly interpolated using the phase variable ϕ:

Density:

$$\rho(\phi) = \frac{1+\phi}{2}\rho_{\text{salt}} + \frac{1-\phi}{2}\rho_{\text{gas}} \tag{11}$$

Viscosity:

$$\mu(\phi) = \frac{1+\phi}{2}\mu_{\text{salt}} + \frac{1-\phi}{2}\mu_{\text{gas}} \tag{12}$$

2.2.3. Summary of the Coupled System

The coupled system governing molten salt infiltration is summarized as:

$$\frac{\partial\phi}{\partial t} + \mathbf{u}\cdot\nabla\phi = \frac{\gamma\lambda}{\epsilon^2}\nabla^2\psi \tag{13}$$

$$\psi = -\epsilon^2\nabla^2\phi + (\phi^2 - 1)\phi \tag{14}$$

$$\rho(\phi)\left(\frac{\partial\mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u}\right) = -\nabla p + \nabla\cdot\tau + G\nabla\phi + \rho(\phi)\mathbf{g} \tag{15}$$

$$\nabla\cdot\mathbf{u} = 0 \tag{16}$$

The variables ϕ, ψ, $\mathbf{u}$, and p are the unknowns.

2.2.4. Boundary Conditions

A boundary condition (BC) is essential for realistically capturing the interaction between the fluids and the graphite surface. These applied BCs are given below:

- **No slip at the solid walls**, which is enforced by:

$$\mathbf{u} = 0 \tag{17}$$

- **No flux through the solid wall**, as given by:

$$\mathbf{n} \cdot \nabla G = 0$$

As $G = \frac{\lambda}{\epsilon^2}\psi$, the above equation can be rewritten as:

$$\mathbf{n} \cdot \nabla \psi = 0 \tag{18}$$

- **The contact angle condition** is enforced to ensure accurate wetting behavior. This BC describes the contact angle of the molten salt at the interface with the graphite surface:

$$\nabla \phi \cdot \mathbf{n} = \frac{1}{\lambda}\frac{3}{4}\sigma \cos(\theta_s)(1-\phi^2) \tag{19}$$

where θ_s is the static contact angle, and $\mathbf{n}$ is the normal vector to the solid wall.

2.3. Finite Element Formulation and Software Implementation

The governing equations (13)–(16) and BCs (17)–(19) are solved within a finite element framework. To facilitate this, the weak forms of their equations are derived, and the newly introduced terms are implemented within the Idaho National Laboratory (INL)'s Multiphysics Object Oriented Simulation Environment (MOOSE) framework [10]. Notably, several terms are already available in MOOSE, thus streamlining the implementation process. MOOSE is inherently designed to simplify the incorporation of new weak forms.

To that end, the weak forms of the governing equations are derived with the test functions $\tilde{\phi}$, $\tilde{\psi}$, $\tilde{\mathbf{u}}$, and $\tilde{p}$, for the variables ϕ, ψ, $\mathbf{u}$, and p, respectively.

For ϕ:

$$\int_\Omega \left(\frac{\partial \phi}{\partial t} + \mathbf{u} \cdot \nabla \phi - \frac{\gamma\lambda}{\epsilon^2}\nabla^2 \psi \right) \tilde{\phi}\, d\Omega \tag{20}$$

$$\int_\Omega \frac{\partial \phi}{\partial t}\tilde{\phi}\, d\Omega + \int_\Omega \mathbf{u} \cdot \nabla \phi \tilde{\phi}\, d\Omega - \frac{\nu\lambda}{\epsilon^2}\int_\Omega \nabla^2 \psi \tilde{\phi}\, d\Omega \tag{21}$$

Expanding the last term using vector calculus identity and the Gauss theorem provides:

$$\int_\Omega \frac{\partial \phi}{\partial t}\tilde{\phi}\, d\Omega + \int_\Omega \mathbf{u} \cdot \nabla \phi \tilde{\phi}\, d\Omega - \frac{\gamma\lambda}{\epsilon^2}\int_{\partial\Omega} \nabla \psi \cdot \mathbf{n}\tilde{\phi}\, d\Gamma - \frac{\gamma\lambda}{\epsilon^2}\int_\Omega \nabla \psi \cdot \nabla \tilde{\phi}\, d\Omega \tag{22}$$

And applying the BCs from Equation 18 results in:

$$\int_\Omega \frac{\partial \phi}{\partial t}\tilde{\phi}\, d\Omega + \int_\Omega \mathbf{u} \cdot \nabla \phi \tilde{\phi}\, d\Omega - \frac{\gamma\lambda}{\epsilon^2}\int_\Omega \nabla \psi \cdot \nabla \tilde{\phi}\, d\Omega \tag{23}$$

Each individual term contributes to the overall residual of the problem and programmed as objects within MOOSE in C++ language. The first term in Equation 23 is already present in the MOOSE system as `ADTimeDerivative`. Subsequently, the following two terms are written as new MOOSE objects, namely, `PhaseFieldAdvection` and `ADPrefactorLaplacianSplit`. Appendix A contains the full MOOSE input file. The following code snippet demonstrates the residual computation for the `PhaseFieldAdvection` object at a quadrature point:

```
ADReal
PhaseFieldAdvection::precomputeQpResidual()
{
  return _velocity[_qp] * _grad_u[_qp];
}
```

Although the term of the form $\int_\Omega \nabla\psi \cdot \nabla\tilde{\phi}$ is present in the existing MOOSE code base, a new object had to be created to account for the prefactor. The following code snippet provides the `PrefactorLaplacianSplit`:

```
ADRealGradient
ADPrefactorLaplacianSplit::precomputeQpResidual()
{
  return (_prefactor/_rho_val[_qp]) *
         ADLaplacianSplit::precomputeQpResidual();
}
```

For ψ:

$$\int_\Omega (\psi + \epsilon^2 \nabla^2 \phi - (\phi^2 - 1)\phi)\tilde{\psi}\, d\Omega \tag{24}$$

$$\int_\Omega \psi\tilde{\psi}\, d\Omega + \epsilon^2 \int_\Omega \nabla^2\phi\tilde{\psi}\, d\Omega - \int_\Omega \phi(\phi^2 - 1)\tilde{\psi}\, d\Omega = 0 \tag{25}$$

$$\int_\Omega \psi\tilde{\psi}\, d\Omega + \epsilon^2 \int_{\partial\Omega} \nabla\phi \cdot \mathbf{n}\tilde{\psi}\, d\Gamma - \epsilon^2 \int_\Omega \nabla\phi \cdot \nabla\tilde{\psi}\, d\Omega - \int_\Omega \phi(\phi^2 - 1)\tilde{\psi}\, d\Omega = 0 \tag{26}$$

Substituting the BC from Equation 19 results in:

$$\int_\Omega \psi\tilde{\psi}\, d\Omega + \epsilon^2 \int_{\partial\Omega} (\frac{1}{\lambda}\frac{3}{4}\sigma cos\theta_s(1 - \phi^2))\tilde{\psi}\, d\Gamma - \epsilon^2 \int_\Omega \nabla\phi \cdot \nabla\tilde{\psi}\, d\Omega - \int_\Omega \phi(\phi^2 - 1)\tilde{\psi}\, d\Omega = 0 \tag{27}$$

As shown here, the first and last terms are already present in the MOOSE system as `ADReaction` and `PhaseFieldCoupledDoubleWellPotential`, respectively. As discussed above, the third term is defined as `ADPrefactorLaplacianSplit`, while the second is the contact angle BC between the fluid and the pore walls. This is implemented as `PhaseFieldContactAngleBC`, and the code snippet is as follows:

```
ADReal
PhaseFieldContactAngleBC::computeQpResidual()
{
  return _test[_i][_qp] * (0.75 * _epsilon * _epsilon / _lambda)
         * _sigma * std::cos(_contactangle) * (1 - _pf[_qp] * _pf[_qp]);
}
```

For $\mathbf{u}$:

$$\int_\Omega \left[\rho(\phi)\left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla\mathbf{u}\right) + \nabla p - \nabla \cdot \tau - G\nabla\phi - \rho(\phi)\mathbf{g}\right] \tilde{\mathbf{u}}\, d\Omega \tag{28}$$

Substituting G in terms of ψ and simplifying provides:

$$\int_\Omega \left[\rho(\phi)\left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u}\right) + \nabla p - \nabla\cdot\tau - \rho(\phi)\mathbf{g}\right]\tilde{\mathbf{u}}\, d\Omega - \frac{\lambda}{\epsilon^2}\int_\Omega \psi\nabla\phi\tilde{\mathbf{u}}\, d\Omega \tag{29}$$

Most of the terms, except the last, are a part of the standard MOOSE Navier-Stokes Module. The last term is implemented as `PhaseFieldTwoPhaseSurfaceTension`, and the code snippet is provided below:

```
ADRealVectorValue
PhaseFieldTwoPhaseSurfaceTension::precomputeQpResidual()
{
  return -_coeff*_auxpf[_qp]*_grad_pf[_qp];
}
```

The newly developed codes, now in the testing phase, are scheduled for integration into the MOOSE framework by August 2025.

2.4. Model Verification

Model verification is crucial to ensure that the governing equations correctly represent the physical behavior, and that the equations are accurately coded within the MOOSE framework. This process involves testing multiple aspects of the model, including the correctness of the BCs and validation against the analytical solutions or the benchmark cases. This ensures robustness and reliability of the model in capturing the desired physical phenomenon.

One of the initial verification steps focuses on validating the formulation and implementation of the contact angle BC. To that end, an oil-water system is chosen, and the simulation set-up closely follows the process described in Zhu et al. [11]. A two-dimensional (2D) rectangular channel with 1.3 m $\times$0.2 m dimensions is modeled, which contains water and oil, as shown in Figure 1(a). The inlet is shown on the left, while the outlet is shown on the right. The top and bottom edges are solid walls, which prevent fluid penetration. The properties for the oil and water are provided in Table 1.

Table 1. Properties of oil and water.

Phase	**Density (kg/m³)**	**Viscosity (Pa·s)**
Water	1000	1×10^{-3}
Oil	840	7.6×10^{-3}

Two different contact angles measuring 30°and 60°, respectively, were initialized to examine the accuracy of the contact angle BC. The simulations were performed, and the contact angles were manually measured from the visualization of the solution. These contact angles agreed with the inputs, thus confirming the accuracy of the contact angle BC formulation and implementation. To further streamline the process, a post-processor to measure the contact angle was coded into the MOOSE framework. The relevant code snippet is provided below:

```
void
ObtainAvgContactAngle::execute()
{
```

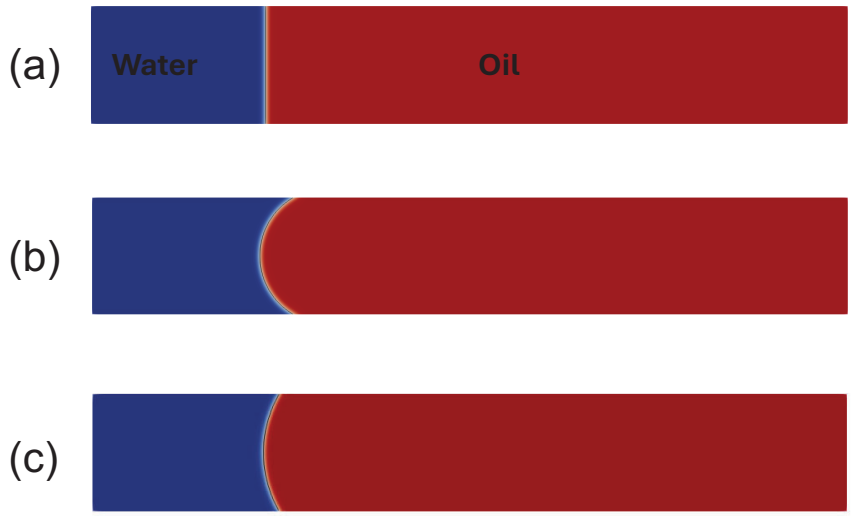


Figure 1. (a) Initial simulation set-up of a 2D rectangular channel containing oil (red region) and water (blue region). Simulation results showing interface configurations with contact angles of (b) 30°and (c) 60°.

```
  for (unsigned int qp = 0; qp < _qrule->n_points(); qp++)
  {
    if (std::abs(_pf[qp]) < 1.0) //Operating only within the interface
    {
      Real tol_val = libMesh::TOLERANCE * libMesh::TOLERANCE;
      const Real weight = _grad_pf[qp].norm();
      _cos_theta_val +=
          _grad_pf[qp] * _normals[qp];
      _total_weight += weight;
    }
  }
}

void
ObtainAvgContactAngle::threadJoin(const UserObject & y)
{
  const ObtainAvgContactAngle &
        pps = static_cast<const ObtainAvgContactAngle &>(y);
  _cos_theta_val += pps._cos_theta_val;
  _total_weight += pps._total_weight;
}

void
ObtainAvgContactAngle::finalize()
{
  gatherSum(_cos_theta_val);
  gatherSum(_total_weight);
  _contact_angle = std::acos(_cos_theta_val / _total_weight)
```

```
                * 180 / libMesh::pi;
}
```

A second verification case focuses on the physical behavior of a two-phase immiscible system, where water (a wetting type fluid) is expected to displace oil (a non-wetting type fluid) due to capillary action. With a zero pressure differential between the inlet and the outlet, the interface is expected to move towards the right. The simulation is set-up with a contact angle of 30°, and the time evolution for the system is presented through snapshots at different time intervals. As observed in Figure 2, the simulation results qualitatively verify the model by demonstrating the expected behavior of the two-phase immiscible system.

Figure 2. Time evolution of the oil-water system at different time intervals at a zero pressure differential.

Finally, for the same system, the displacement of the interface with time from the simulation is compared to the analytical solution [11]. This comparison is shown in Figure 3, which illustrates that the simulation results are in excellent agreement with the analytical solution.

This verification process demonstrates that the model accurately captures the expected physical behavior of two-phase immiscible fluids on a simple rectangular geometry, establishing confidence in this model.

2.5. Application to IG-110 Graphite

Following the above verification, the model then is used to simulate a more realistic scenario, namely, FLiNaK (a molten salt) infiltrating into the pore microstructure of IG-110 grade nuclear graphite. A computed tomography (CT) image of a slice of IG-110 was obtained from Oak Ridge National Laboratory (ORNL). The CT slice has dimensions 200 μm $\times$200 μm, as shown in Figure 4. Extensive preprocessing was performed using MOOSE and Sandia National Laboratory's CUBIT [12] software to smooth the sharp edges of the pore contours, define the boundaries, extract the open pore channel, and mesh the geometry.

The extracted open pore is used for the FLiNaK infiltration simulation. The initial simulation setup is shown in Figure 5(a), illustrating the inlet and outlet channels for the fluids, as indicated by the arrows. Most of the pore is already occupied by argon gas, indicated by the red-colored region in Figure 5(a). All surfaces, except for the inlets and outlets, are solid walls that prevent fluid infiltration. The simulation is performed with a pressure differential of 1 atm between the inlet and outlet channels. A snapshot in time of the system is

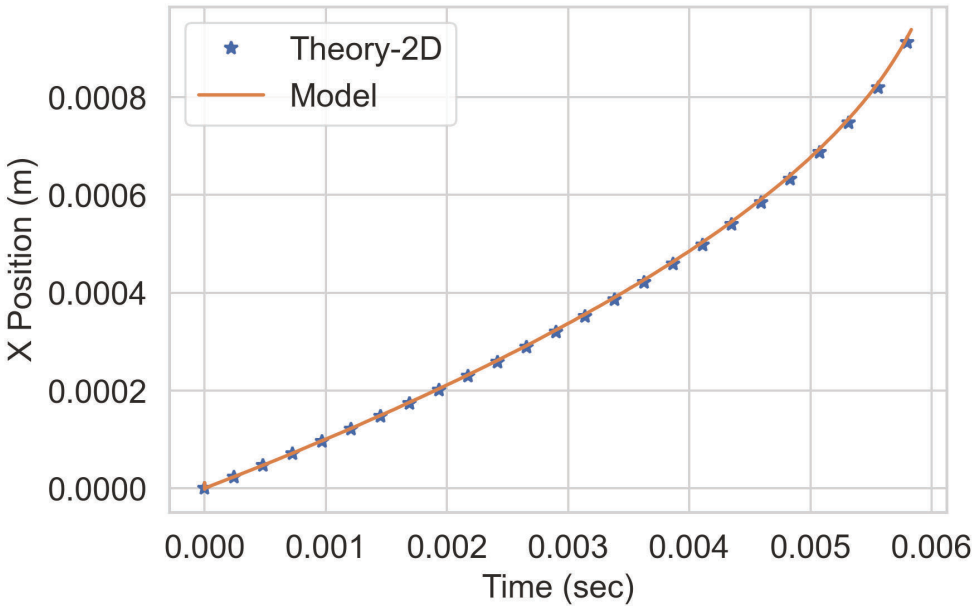


Figure 3. Verification of evolution of interface displacement with the analytical solution.

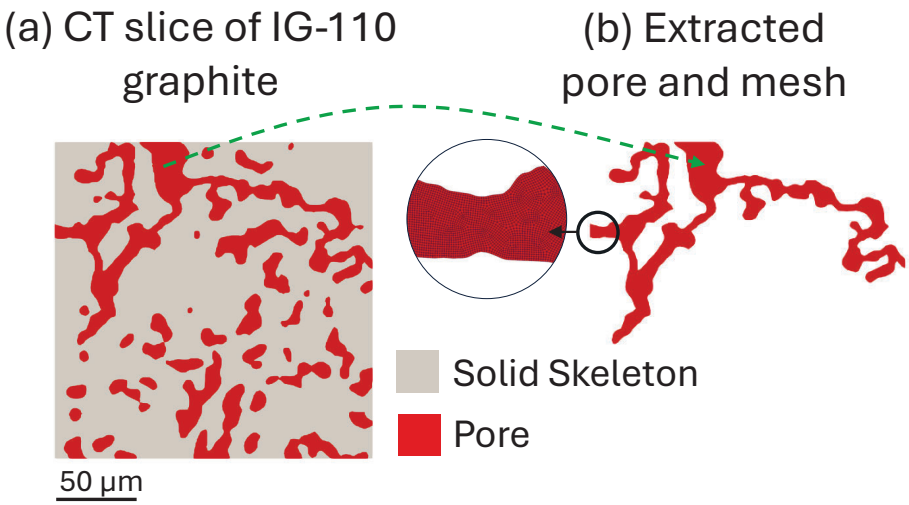


Figure 4. (a) A 2D slice CT scan of an IG110 graphite sample showing the solid skeleton and pores. (b) The extracted pore geometry with the finite element mesh is shown in the inset.

shown in Figure 5(b), depicting that the FLiNaK has infiltrated the pore channel, displacing the argon gas. While the solution is taken at a converged step at a particular time (Figure 5(b)), running the same simulation for a longer duration resulted in the vanishing of the FLiNaK phase at the bottom corner, potentially due to the use of incompressible Navier-Stokes equations in the formulation. Additionally, certain parameter sets led to convergence issues.

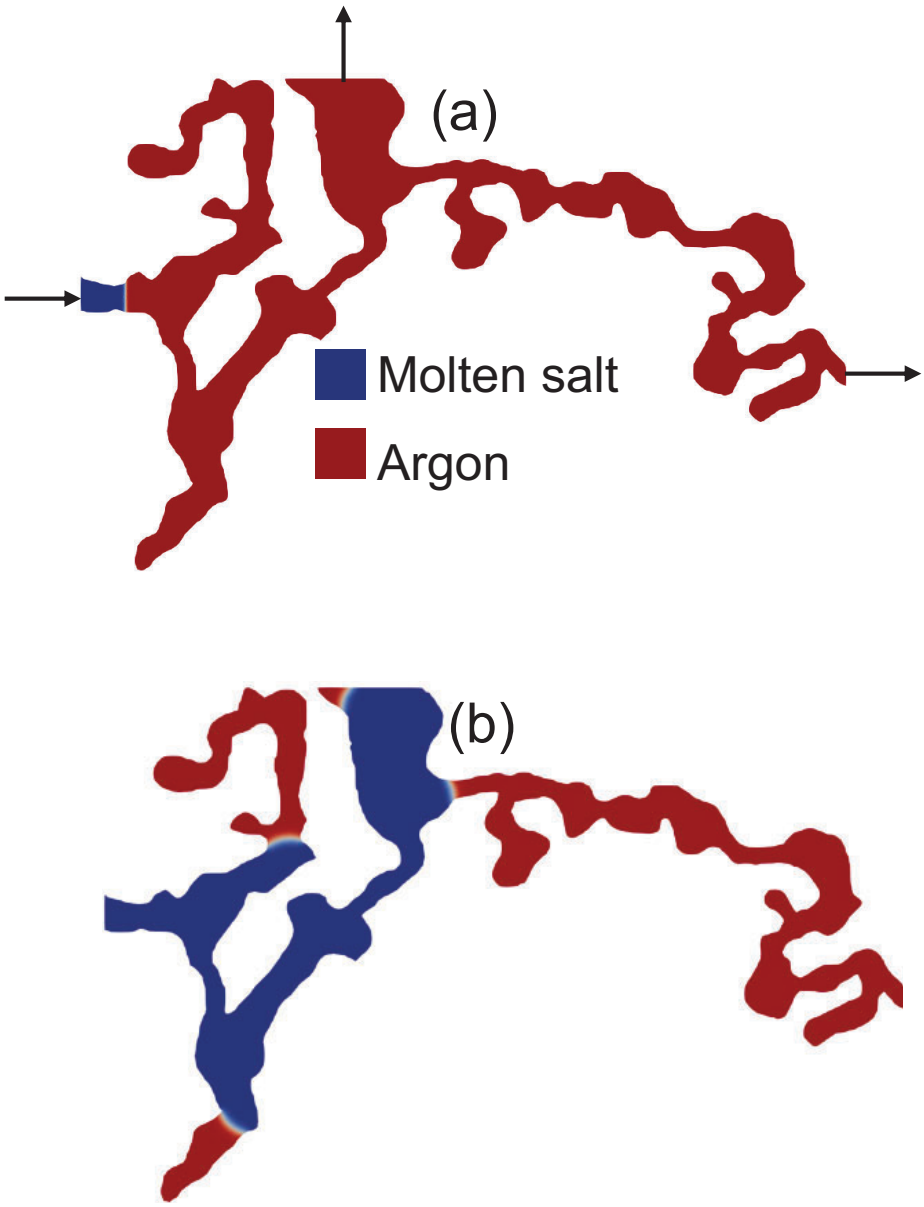


Figure 5. (a) Initial simulation setup of the FLiNaK and Ar system, with the inlets and outlets shown by the arrows. (b) A snapshot in time illustrating the infiltration of FLiNaK into the pores of the IG-110 graphite.

2.6. Future Work

Future work will focus on resolving the non-physical behavior and convergence issues observed in the FLiNaK infiltration simulations, which potentially were caused by using incompressible Navier-Stokes equations. The computational models will be refined for improved stability and accuracy and these models will be validated against the experimental data. Optimization of the infiltration process then will follow by exploring various parameters to enhance efficiency. Once validated, the improved models will be integrated into the MOOSE code base, ensuring thorough testing and documentation.

3. MODEL DEVELOPMENT FOR GRAPHITE OXIDATION

3.1. Physics of Graphite Oxidation

Graphite oxidation in GCRs arises from the interaction of oxygen with the porous microstructure of graphite. This process typically takes place under two primary scenarios: acute oxidation and chronic oxidation. Acute oxidation happens during events such as air or water ingress in accident conditions, potentially leading to rapid oxidation of graphite. Chronic oxidation, on the other hand, occurs due to the presence of trace oxygen impurities (e.g., air and water) within the system, leading to a slower rate of oxidation as compared to the former. In both cases, the oxidation process potentially compromises the structural integrity of the graphite components, which is crucial for the safety and performance of the reactors.

The most serious oxidation phenomenon involves oxygen diffusing into the interior of the graphite through the internal pore microstructure. Nuclear graphite has a 15-20% total porosity level with an open porosity of 8-10% to mitigate irradiation-induced material property changes. Only the open porosity is accessible to oxygen diffusion. Once inside the pore microstructure, oxygen reacts with the carbon at the exposed surfaces leading to the production of carbon monoxide (CO) and carbon dioxide (CO_2), as per the following equation [13]:

$$\mathrm{C} + \left(1 - \frac{x}{2}\right) \mathrm{O_2} \rightarrow x\mathrm{CO} + (1 - x)\mathrm{CO_2} \tag{30}$$

This process leads to loss of carbon from the graphite, thus reducing the bulk density and altering its mechanical and thermal properties. The rate of oxidation and mass loss are primarily governed by the diffusion of the gases, chemical reaction, and heat transfer phenomenon, and are influenced by several factors such as oxygen concentration, pore structure of the graphite, temperature, etc. A computational model incorporating these phenomena is crucial to understand the oxidation behavior of graphite.

Nuclear graphite, as a porous material, contains micron-sized pores, while the reactor components (or even the lab-scale testing specimens) are many orders of magnitude larger compared to the pores. Therefore, explicitly resolving the pore structure in component-level simulations is extremely challenging. To address this, the computational modeling of graphite oxidation follows a two-step approach. First, the mass and heat transport equations are formulated at the pore scale. Then, using the mathematical principles of volume averaging, these equations are upscaled to the macro-scale, applicable for component simulations. The pore-scale transport equations are fundamental governing laws, which are straightforward to express and detailed in the next section.

3.2. Pore-Scale Transport Equations

In this section, both the mass and heat transport equations are provided at the microscale. The mass transfer equations are only valid in the pore domain, while the heat transfer equations are exclusive to the solid skeleton of the graphite. These equations form the foundation for deriving the macroscopic equations.

3.2.1. Mass Transport

The mass transport within the porous graphite involves the diffusion of O_2 through the pore channels, and a subsequent reaction. This subsection details the governing equations and BCs.

3.2.1.1. Governing Equation

For each species i (O_2, CO, CO_2, N_2, Ar), the conservation of mass in the pore space is governed by Fick's second law:

$$\frac{\partial C_i}{\partial t} = \nabla \cdot (D_i \nabla C_i) \tag{31}$$

where:

- C_i: local concentration of species i (mol/m^3)
- D_i: diffusion coefficient of species i in a gas mixture inside the pore space (m^2/s).

The full list of species includes:

- O_2: oxygen
- CO: carbon monoxide
- CO_2: carbon dioxide
- N_2: nitrogen
- Ar: argon.

This equation states that the rate of change of species concentration within the pore is due to diffusive flux. Note that the reaction occurs at the surface of the pore-solid interface, and these must be accounted for in the form of BCs.

3.2.1.2. Boundary Conditions at the Pore-Solid Interface

At the pore-solid interface, the mass flux of a reactive species is proportional to the reaction rate occurring at the surface. The reaction consumes O_2 and produces CO and CO_2. It is assumed that the reaction rate is first-order with respect to the concentration of the O_2 at the interface:

$$R_C = k_{eff} \mathrm{SA} C_{O_2}^{surface} \tag{32}$$

where:

- R_C: local reaction rate at the interface (mol/m^2·s)
- k_{eff}: effective reaction rate constant normalized to active surface area $(1/m^2 s)$
- SA: active surface area (m^2)
- $C_{O_2}^{surface}$: local concentration of O_2 at the interface (mol/m^2).

k_{eff} is a function of surface area.

With the above assumption, the flux boundary conditions at the interface are provided by:

$$-D_{\mathrm{O_2}}\nabla C_{\mathrm{O_2}}\cdot\mathbf{n} = \left(1-\frac{x}{2}\right)R_C = \left(1-\frac{x}{2}\right)k_{eff}\mathrm{SAC}_{\mathrm{O_2}}^{surface} \quad \text{(oxygen)} \tag{33}$$

$$-D_{\mathrm{CO}}\nabla C_{\mathrm{CO}}\cdot\mathbf{n} = -xR_C = -xk_{eff}\mathrm{SAC}_{\mathrm{O_2}}^{surface} \quad \text{(carbon monoxide)} \tag{34}$$

$$-D_{\mathrm{CO_2}}\nabla C_{\mathrm{CO_2}}\cdot\mathbf{n} = -(1-x)R_C = -(1-x)k_{eff}\mathrm{SAC}_{\mathrm{O_2}}^{surface} \quad \text{(carbon dioxide)} \tag{35}$$

These BCs link the diffusive flux of each species to the chemical reaction occurring at the pore surface.

3.2.2. Heat Transfer

The heat transfer equation applies to the solid skeleton of the graphite. The governing equations and BCs are provided below.

3.2.2.1. Governing Equation

The governing equation is a transient heat conduction equation given by:

$$\frac{\partial}{\partial t}(\rho c_p T) = \nabla\cdot(K\nabla T) \tag{36}$$

where:

- ρ: density of the solid graphite (kg/m^3),
- c_p: specific heat capacity of the solid graphite (J/(kg·K)),
- T: temperature (K),
- K: thermal conductivity of the solid graphite (W/(m·K)),
- q_v: volumetric heat source term (W/m^3).

3.2.2.2. Boundary Conditions at the Pore-Solid Interface

The reaction described by Equation 30 happens at the interface, which contributes to heat generation as this is an exothermic reaction. This is accounted via a heat flux BC.

$$-K\nabla T\cdot\mathbf{n} = \Delta H_{\text{reaction}}R_C = \Delta H_{\text{reaction}}k_{eff}\mathrm{SAC}_{\mathrm{O_2}}^{surface} \tag{37}$$

3.3. Review of the Existing Model

To derive the macroscopic model, the equations presented in Section 3.2 must be upscaled using the principles of volume averaging. While a macroscopic model for simulating graphite oxidation already exists, the upscaling process requires further refinement. This section provides a critical review of the existing model.

3.3.1. Mass Transport

For each species (O_2, CO, CO_2, N_2, Ar), the conservation of mass at the macroscale is given as:

For O_2:

$$\frac{\partial(\epsilon C_{O_2})}{\partial t} = \nabla(D_{\text{eff},O_2}\nabla C_{O_2}) - \left(1 - \frac{x}{2}\right) k_{eff}\text{SA}C_{O_2} \tag{38}$$

For CO:

$$\frac{\partial(\epsilon C_{CO})}{\partial t} = \nabla(D_{\text{eff,CO}}\nabla C_{CO}) + x k_{eff}\text{SA}C_{O_2} \tag{39}$$

For CO_2:

$$\frac{\partial(\epsilon C_{CO_2})}{\partial t} = \nabla(D_{\text{eff},CO_2}\nabla C_{CO_2}) + (1-x) k_{eff}\text{SA}C_{O_2} \tag{40}$$

For N_2:

$$\frac{\partial(\epsilon C_{N_2})}{\partial t} = \nabla(D_{\text{eff},N_2}\nabla C_{N_2}) \tag{41}$$

For Ar:

$$\frac{\partial(\epsilon C_{Ar})}{\partial t} = \nabla(D_{\text{eff,Ar}}\nabla C_{Ar}) \tag{42}$$

References in the classical literature on the upscaling process in transport media [14–18] were reviewed to assess this formulation, and a number of issues were observed. From Equations (38)-(42), the time derivative term appears to be properly upscaled. However, the second term (of the form $\nabla(D_{eff,i}\nabla C_i)$) presents two critical issues: First, this should be a flux term represented by a divergence operator ($\nabla \cdot ()$), instead of a gradient operator. Second, upscaling of flux terms results in the introduction of a porosity term (ϵ), which is currently missing. Lastly, the source terms arising from the upscaling of the BCs in Equations (33)-(35) are potentially missing a porosity term (ϵ). In addition, it is important to perform surface-to-volume conversions to ensure consistency throughout the process.

3.3.2. Heat Transport

$$\frac{\partial}{\partial t}(\rho c_p T) = \nabla \cdot (K_{\text{eff}}\nabla T) + \Delta H_{\text{reaction}} k_{\text{eff}} \text{S}_A C_{O_2} \tag{43}$$

The BC in Equation 37 upscales to the source term in the macroscopic equation for heat transfer. However, a porosity term (ϵ) is potentially missing, which could impact the accuracy of the heat transport equation.

3.4. Future Work

To address the inconsistencies identified in the existing model, future work will primarily focus on several key areas. First, the governing equations will be derived using the principles of volume-averaging, with careful attention to handling the porosity terms and surface-to-volume transformations. Next, verification cases will be developed to systematically evaluate the model in collaboration with a subject matter expert in transport in porous media. Following this, the equations will be implemented in the MOOSE framework, emphasizing modularity and readability. Finally, the new model will be validated against in-house experiments conducted at different temperatures, focusing on the evolution of mass loss and local density of graphite.

4. SUMMARY

This report focuses on developing computational models to understand two critical degradation mechanisms in nuclear graphite used in advanced reactor designs: molten salt infiltration in MSRs and oxidation in GCRs. These degradation mechanisms can impact the structural integrity and operational lifetime of the graphite components, which motivates their detailed assessment using modeling and simulation tools.

For molten salt infiltration, the report introduces a computational framework that integrates incompressible Navier-Stokes equations with the phase-field method to simulate molten salt penetration into the interconnected pore structure of the graphite. Initial model verification was carried out using two-phase flows in two dimensions, confirming the model's accuracy in capturing the fundamental physical behaviors and the matching analytical solutions. This framework was then applied to a sample of IG-110 nuclear graphite, utilizing the pore geometry extracted from the CT scans to analyze the infiltration behavior of the FLiNaK molten salt. Verification of the models is a crucial part of the development process. For the molten salt infiltration model, verification involved testing various aspects of the model, including BCs and validation against the analytical solutions or benchmark cases. The successful application of the model to the IG-110 graphite sample demonstrated its capability to capture realistic infiltration behavior, although some non-physical behaviors were observed. These issues will be addressed in future work to enhance the model's robustness and accuracy.

For oxidation, the report discusses the pore-scale mass and heat transport models that incorporate diffusion of gases, reaction kinetics, and thermal effects. A critical review of the existing macroscopic oxidation model revealed inconsistencies, particularly in the upscaling of reaction kinetics, flux terms, and surface-to-volume transformations. These inconsistencies suggest that current formulations may not accurately capture the oxidation process of graphite, highlighting the need for refined model development. Future work will focus on rigorously deriving the upscaled equations using volume-averaging principles, potentially in collaboration with subject matter experts, to ensure accurate and reliable predictions.

Future efforts focused on oxidation will concentrate on deriving the governing equations using volume-averaging principles, with particular attention to correctly handling porosity terms and surface-to-volume transformations. Verification cases will be developed to systematically evaluate the new model, which will then be implemented in the MOOSE framework. The improved model will be validated against experimental data obtained from in-house experiments conducted at various temperatures, focusing on the evolution of mass loss and changes in the local density of the graphite.

In summary, the work documented here advances the development of physics-based computational models for graphite degradation, providing improved predictive capabilities for next-generation nuclear reactor designs. The findings and insights from these models will significantly aid in the design and performance analysis of graphite components, ensuring reactor safety, structural integrity, and extended operational lifetimes.

5. REFERENCES

[1] V. Prithivirajan, "Examining graphite degradation in molten salt environments: A chemical, physical, and material analysis," Tech. Rep. INL/RPT-23-75892, Idaho National Laboratory (INL), Idaho Falls, ID (United States), August 2024.

[2] B. J. Marsden, G. N. Hall, and A. N. Jones, "4.13 - graphite in gas-cooled reactors ," in *Comprehensive Nuclear Materials (Second Edition)*, pp. 357–421, Oxford: Elsevier, second edition ed., 2020.

[3] J. Standring, "Calculation of the graphite weight-loss in civil magnox and advanced gas-cooled reactors," *Journal of Nuclear Energy. Parts A/B. Reactor Science and Technology*, vol. 20, no. 3, pp. 201–217, 1966.

[4] W. Windes, G. Strydom, R. Smith, and J. Kane, "Role of nuclear grade graphite in controlling oxidation in modular HTGRs," Tech. Rep. INL/EXT-14-31720, Idaho National Laboratory (INL), Idaho Falls, ID (United States), 2014.

[5] R. C. Robertson, "MSRE design and operations report, part i: Description of reactor design," Tech. Rep. ORNL-TM-0728, Oak Ridge National Laboratory (ORNL), Oak Ridge, TN (United States), January 1965.

[6] J. Bass, "Graphite degradation modeling and analysis," Tech. Rep. TLR-RES/DE/REB-2022-11, INL/EXT-22-65240, Idaho National Laboratory, 2022.

[7] P. Yue, C. Zhou, J. J. Feng, C. F. Ollivier-Gooch, and H. H. Hu, "Phase-field simulations of interfacial dynamics in viscoelastic fluids using finite elements with adaptive meshing," *Journal of Computational Physics*, vol. 219, no. 1, pp. 47–67, 2006.

[8] C. Zhou, P. Yue, J. J. Feng, C. F. Ollivier-Gooch, and H. H. Hu, "3d phase-field simulations of interfacial dynamics in newtonian and viscoelastic fluids," *Journal of Computational Physics*, vol. 229, no. 2, pp. 498–511, 2010.

[9] J. W. Cahn and J. E. Hilliard, "Free energy of a nonuniform system. i. interfacial free energy," *The Journal of Chemical Physics*, vol. 28, no. 2, pp. 258–267, 1958.

[10] C. J. Permann, D. R. Gaston, D. Andrš, R. W. Carlsen, F. Kong, A. D. Lindsay, J. M. Miller, J. W. Peterson, A. E. Slaughter, R. H. Stogner, and R. C. Martineau, "MOOSE: Enabling massively parallel multiphysics simulation," *SoftwareX*, vol. 11, p. 100430, 2020.

[11] Z. Zhu, Z. Song, Z. Shao, M. Wu, and X. Xu, "Simulation of imbibition in porous media with a tree-shaped fracture following the level-set method," *Physics of Fluids*, vol. 33, p. 082109, August 2021.

[12] C. Stimpson, G. Sjaardema, C. Dudley, R. Yorgason, B. Hanks, D. White, R. Kerr, S. Mitchell, J. Shepherd, T. Tautges, *et al.*, "Cubit v.16.x," August 2022.

[13] J. J. Kane, C. I. Contescu, R. E. Smith, G. Strydom, and W. E. Windes, "Understanding the reaction of nuclear graphite with molecular oxygen: Kinetics, transport, and structural evolution," *Journal of Nuclear Materials*, vol. 493, pp. 343–367, 9 2017.

[14] S. Whitaker, "The transport equations for multi-phase systems," *Chemical Engineering Science*, vol. 28, no. 1, pp. 139–147, 1973.

[15] S. Whitaker, "A simple geometrical derivation of the spatial averaging theorem," *Chemical Engineering Education*, vol. 19, no. 1, pp. 18–52, 1985.

[16] W. G. Gray, "A derivation of the equations for multi-phase transport," *Chemical Engineering Science*, vol. 30, no. 2, pp. 229–233, 1975.

[17] H. D. Lugo-Méndez, F. J. Valdés-Parada, M. L. Porter, B. D. Wood, and J. A. Ochoa-Tapia, "Upscaling diffusion and nonlinear reactive mass transport in homogeneous porous media," *Transport in Porous Media*, vol. 107, pp. 683–716, 2015.

[18] B. Goyeau, "Macroscopic conduction models by volume averaging for two-phase systems," in *Thermal Nanosystems and Nanomaterials*, pp. 95–105, Springer, 2009.

A. APPENDIX A

The MOOSE input file used for model verification is provided below:

```
sigma = 25e-3 #surface tension coefficient
epsilon = 4e-6 #width parameter
nu = 1e-4 #mobility parameter
contactangle = 2.61799#$2.0944##0.523599#1.0472
lambda = ${fparse 3*sigma*epsilon/(2*sqrt(2))}
prefactor_phi = ${fparse nu*lambda/(epsilon*epsilon)}
prefactor_psi = ${fparse -epsilon*epsilon}
coeff = ${fparse lambda/(epsilon*epsilon)}

[Mesh]
  [gen]
    type = GeneratedMeshGenerator
    dim = 2
    xmin = 0
    xmax = 1.3e-3
    ymin = 0
    ymax = 0.2e-3
    nx = 260
    ny = 40
    elem_type = QUAD9
  []
[]

[GlobalParams]
  gravity = '0 -9.8 0'
[]

[ICs]
  [pf_ic]
    type = BoundingBoxIC
    variable = pf
    x1 = 0.3e-3
    y1 = -0.1e-3
    x2 = 1.4e-3
    y2 = 0.3e-3
    inside = 1
    outside = -1
    int_width = ${fparse 2*sqrt(2)*epsilon}
  []
  [velocity]
    type = VectorConstantIC
    x_value = 0.0
    y_value = 0.0
    variable = velocity
  []
[]


[Variables]
  [pf]
```

```
    family = LAGRANGE
    order = second
  []
  [auxpf]
    family = LAGRANGE
    order = second
  []
  [velocity]
    family = LAGRANGE_VEC
  []
  [p]
  []
[]

[Kernels]
  [mass]
    type = INSADMass
    variable = p
  []
  [mass_pspg]
    type = INSADMassPSPG
    variable = p
  []

  [momentum_time]
    type = INSADMomentumTimeDerivative
    variable = velocity
  []

  [momentum_convection]
    type = INSADMomentumAdvection
    variable = velocity
  []

  [momentum_viscous]
    type = INSADMomentumViscous
    variable = velocity
    viscous_form = 'traction'
    mu_name = 'mu'
  []

  [momentum_pressure]
    type = INSADMomentumPressure
    variable = velocity
    pressure = p
    integrate_p_by_parts = false
  []

  [gravity]
    type = INSADGravityForce
    variable = velocity
  []

  [momentum_supg]
```

```
  type = INSADMomentumSUPG
  variable = velocity
  velocity = velocity
[]

[surface_tension]
  type = PhaseFieldTwoPhaseSurfaceTension
  variable = velocity
  pf = pf
  auxpf = auxpf
  coeff = ${coeff}
[]

[phasefield_timederivative]
  type = ADTimeDerivative
  variable = pf
[]

[phasefield_supg]
  type = PhaseFieldTimeDerivativeSUPG
  velocity = velocity
  variable = pf
[]

[phasefield_advection]
  type = PhaseFieldAdvection
  velocity = velocity
  variable = pf
[]

[phasefield_advection_supg]
  type = PhaseFieldAdvectionSUPG
  velocity = velocity
  variable = pf
[]

[phasefield_laplacian]
  type=ADPrefactorLaplacianSplit
  variable = pf
  c = auxpf
  prefactor = ${prefactor_phi}
[]

[Auxphasefield_firstorder]
  type=ADReaction
  variable = auxpf
  rate = 1.0
[]

[Auxphasefield_laplacian]
  type=ADPrefactorLaplacianSplit
  variable = auxpf
  c = pf
  prefactor=${prefactor_psi}
```

```
  []

  [Auxphasefield_doublewell]
    type=PhaseFieldCoupledDoubleWellPotential
    variable = auxpf
    c = pf
    prefactor=-1.0
  []
[]

[BCs]
  [no_slip]
    type = VectorFunctionDirichletBC
    variable = velocity
    boundary = 'top bottom'
  []
  [pressure_L]
    type = DirichletBC
    variable = p
    boundary = 'left'
    value = 0.0
  []

  [pressure_R]
    type = DirichletBC
    variable = p
    boundary = 'right'
    value = 0.0
  []

  [ContactangleBC]
    type=PhaseFieldContactAngleBC
    variable = auxpf
    pf = pf
    epsilon = ${epsilon}
    lambda=${lambda}
    sigma=${sigma}
    contactangle=${contactangle}
    boundary = 'top bottom'
  []
[]

[Materials]
  [rho]
    type = PhaseFieldTwoPhaseMaterial
    prop_name = rho
    prop_value_1 = 1000
    prop_value_2 = 840
    pf = pf
    outputs = exodus
  []
  [mu]
    type = PhaseFieldTwoPhaseMaterial
    prop_name = mu
```

```
    prop_value_1 = 1e-3
    prop_value_2 = 7.6e-3
    pf = pf
    outputs = exodus
  []

  [ins_mat]
    type = INSADTauMaterial
    velocity = velocity
    pressure = p
    alpha = .1
  []

[]

[Postprocessors]
  [contact_angle_top]
    type = ObtainAvgContactAngle
    boundary = bottom
    pf=pf
    execute_on = 'timestep_end'
  []
  [x_position]
    type = FindValueOnLine
    start_point = '0 0.0001 0'
    end_point = '0.0013 0.0001 0'
    v = pf
    target = 0.0
    tol = 1e-6
  []
[]

[Preconditioning]
  [SMP]
    type = SMP
    full = true
  []
[]

[Executioner]
  type = Transient
  solve_type = NEWTON
  start_time = 0
  end_time = 8e-3
  dtmax = 0.25
  [TimeStepper]
    type = IterationAdaptiveDT
    dt = 1e-10
    iteration_window = 2
    optimal_iterations = 10
    growth_factor = 2
    cutback_factor = 0.5
  []
  petsc_options_iname = '-pc_type'
```

```
  petsc_options_value = 'lu       '
  line_search = 'none'
  nl_rel_tol = 1e-10
  nl_abs_tol = 1e-12
  nl_max_its = 20
  nl_forced_its = 3
  l_tol = 1e-6
  l_max_its = 20
[]

[Outputs]
  exodus = true
  [csv]
    type = CSV
    sync_times = '0 1e-6 0.001 0.002 0.003 0.004 0.005 0.006'
    sync_only = true
    time_data = true
  []
[]
```