Creep and Creep Fracture Modeling with Surrogate Creep Models and the Extended Finite Element Method

INL/RPT-24-82173

Scientific Discovery through Advanced Computation

SEPTEMBER 2024

K M Zaheen Nasir and Wen Jiang

North Carolina State University

Benjamin W. Spencer

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-24-82173

Creep and Creep Fracture Modeling with Surrogate Creep Models and the Extended Finite Element Method

K M Zaheen Nasir and
Wen Jiang
North Carolina State University
Benjamin W. Spencer
Idaho National Laboratory

September 2024

Idaho National Laboratory
Computational Mechanics & Materials
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

ABSTRACT

Alloy components in advanced nuclear reactors will be subjected to environmental conditions that could include high temperatures, irradiation, and exposure to corrosive salts. These conditions could lead to the formation of crack-like defects, which could grow over time in a mechanism known as creep crack growth (CCG). Predicting growth rates of these defects is important for assessing the safe operating life of advanced reactors. This project documents progress toward developing and testing next-generation data-driven constitutive models for deformation creep. It also documents the application of the extended finite element method in conjunction with surrogate creep models to predict CCG parameters under a variety of conditions. These important incremental developments contribute to the longer-term objective of developing microstructure-aware constitutive models that can be used for predicting creep deformation and CCG at the component scale with improved accuracy.

ACKNOWLEDGMENTS

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 "Simulation of the Response of Structural Metals in Molten Salt Environment" project under the US Department of Energy's Scientific Discovery through Advanced Computing (SciDAC) program.

This research made use of the resources of the High Performance Computing Center at Idaho National Laboratory, 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

CCG	creep crack growth
FE	finite element
INL	Idaho National Laboratory
ML	machine learning
MoE	Mixture of Experts
MOOSE	Multiphysics Object-Oriented Simulation Environment
NCSU	North Carolina State University
ROM	reduced order model
SM	surrogate model
XFEM	extended finite element method

Page intentionally left blank

Creep and Creep Fracture Modeling with Surrogate Creep Models and the Extended Finite Element Method

1. INTRODUCTION

Alloys used for the components in molten-salt reactors must be able to perform acceptably in the reactor environment, which will include exposure to salt, high temperature, and irradiation. To design reactor components, it is essential to be able to predict the deformation creep response. In addition, there is the potential for crack defects to initiate and propagate in those components under service conditions. Crack growth can occur under steady-state conditions or during cyclic loading in a mechanism known as creep-fatigue interaction. In this report, crack growth affected by creep deformation ahead of the crack tip is referred to as creep crack growth (CCG).

The Simulation of the Response of Structural Metals in Molten Salt Environment project under the US Department of Energy's Scientific Discovery through Advanced Computing (SciDAC) program is developing capabilities for predicting multiple degradation mechanisms—including corrosion, creep, and CCG—for salt-facing alloys relevant to advanced reactors. Idaho National Laboratory (INL) and North Carolina State University (NCSU) are supporting this effort by testing and integrating surrogate creep-deformation models into finite element (FE) codes and developing capabilities for CCG modeling that make use of these surrogate models (SMs). This report documents efforts of INL and NCSU in both of those areas to support that project.

Creep deformation in alloys is highly dependent on microstructure, processing, and environmental conditions. High-fidelity mesoscale models that explicitly represent the crystallographic structure of alloys can predict many of these effects. These models are computationally expensive, and because they represent material at a scale much smaller than that of engineering components, deploying them to predict component response is challenging. SMs that represent an alloy's deformation response based on data obtained from mesoscale simulations are being developed by other partner institutions in the current SciDAC project. Polynomial-based SMs have been previously developed in other programs, and the current SciDAC project is exploring the use of machine learning (ML) methods as an alternative surrogate-modeling method. NCSU has developed a Python-based test bed for these models to rapidly assess their robustness and accuracy before deploying them in a large-scale FE code, as described in Section 2.

Creep deformation is an important driver of CCG, so it is natural to take advantage of the developments in SMs for engineering-scale component response to predict CCG. The other aspect of the work, documented in Section 3, is an effort by INL to deploy a surrogate creep model to capture the deformation response ahead of a crack tip under high-temperature conditions. The extended finite element method (XFEM) is a powerful technique that can be used to model the propagation of discrete cracks in an FE mesh independently of the FE discretization. This report documents efforts to assess using XFEM together with surrogate creep models for CCG modeling within the Multiphysics Object-Oriented Simulation Environment (MOOSE) framework, which is the target platform for deploying the capabilities described here for large-scale simulations.

2. CREEP SURROGATE MODELING

Using data obtained from viscoplastic self-consistent models of material response at the mesoscale, Sandia National Laboratories and Los Alamos National Laboratory jointly developed a Mixture of Experts (MoE) ML model to predict the elastic-viscoplastic response of HT-9 steel under reactor operational conditions. To deploy this ML model, it needs to be faithfully integrated within an FE framework to leverage its superior prediction ability for analyzing versatile and complex geometric entities. The stand-alone Python-based finite element code PyFEM was chosen as the initial framework for model integration and testing. This report presents the work done to date and its findings. The description of this effort is organized as follows: Section 2.1 describes the PyFEM architecture and the details of the implementation. Section 2.2 describes the numerical verification of the accurate integration. Findings from several investigations are introduced in Section 2.3. Section 2.4 discusses the results of a few simulations. Section 2.5 presents the strain rate map generated using varying stress states and temperatures. Finally, Section 2.6 provides concluding remarks along with the limitations and future work.

2.1. Implementation Details

2.1.1. PyFEM Architecture

PyFEM simulations are executed by invoking the `PyFEM.py` Python script from the terminal, with the name of an input file specified as an command-line argument to that script. Depending on the operating system used, a shell script can also be used to run the PyFEM simulation. The PyFEM input file is a project file with the `*.pro` extension, which is a text file with material, solver, input, and output specifications that can be written using any text or code editors. The `*.pro` file is linked to a `*.dat` file that contains all the meshing information along with the boundary conditions. The input-file parser goes through the `*.pro` file and instantiates the needed solver and material objects among the various modules required for the FE analysis of the given geometry. At each quadrature point the `getStress` method of that respective material model is called to calculate the stress from the updated kinematics. Therefore, the implementation of a new constitutive model necessitates the redefinition of these material classes along with their member variables and methods.

2.1.2. Viscoplasticity Model (Power-Law Creep)

As an initial step toward integrating the MoE elasto-viscoplasticity model into PyFEM, a power-law viscoplasticity model was implemented within the PyFEM framework. This power-law viscoplasticity model computes the effective creep strain $\dot{\varepsilon_p}$ as:

$$\dot{\varepsilon_p} = A\,(\sigma_e^{tr} - 3G\Delta p)^n\,(t - t_o)^m \tag{1}$$

where G is the isotropic shear modulus and A is a material-dependent constant that can be easily extended to useful quantities. For example, one can express A as $A = e^{(\frac{-Q}{RT})}$. Here, Q is the activation energy, R is the universal gas constant, T is the temperature, t and t_o are current and initial times, respectively, and n and m are exponent values. This power-law creep model is implicitly integrated by evaluating it within the radial-return method (Section 2.1.3) as shown in Algorithm 1, following Dunne and Petrinić [1].

Algorithm 1 Return Mapping Implementation (Power-Law Creep)

Initialize $residual \leftarrow 0.0$
Initialize $k \leftarrow 0$
Define $tolerance$ as the convergence threshold (small value)
Define $max\_iterations \leftarrow 100$
Define $deqpl$ as the effective plastic strain increment
Define other material and problem parameters (A, $3G$, n, $(t - t_o)^m$, Δt, σ_e)
while $|residual| > tolerance$ or $k == 0$ **do**
    $k \leftarrow k + 1$
    if $k > max\_iterations$ **then**
        break ▷ Exit the loop if iteration limit is reached
    end if
    Compute $residual \leftarrow A \times (\sigma_e - 3G \times deqpl)^n \times (t - t_o)^m \times \Delta t - deqpl$
    if $|residual| < tolerance$ **then**
        exit ▷ Convergence achieved
    end if
    Compute $jacobian \leftarrow -A \times 3G \times n \times (\sigma_e - 3G \times deqpl)^{n-1} \times (t - t_o)^m \times \Delta t - 1.0$
    Update $deqpl \leftarrow deqpl - \frac{residual}{jacobian}$ ▷ Newton-Raphson update
end while

2.1.3. Equations and Algorithms for MoE Model Integration

The mechanical analysis of arbitrary model geometries requires obtaining the stresses and strains derived from the displacement field associated with that entity given a particular set of boundary conditions. Regardless of the complexity of the problem, the displacement field is obtained by solving a force balance equation. Equation 2 represents the discretized nonlinear equilibrium equation of FE to be solved to determine the displacement **u** by the FE framework [1].

$$\boldsymbol{F^{inertial}(u)} + \boldsymbol{F^{int}(u)} - \boldsymbol{F^{ext}(u)} = 0 \tag{2}$$

Here, $\boldsymbol{F^{inertial}(u)}$, $\boldsymbol{F^{int}(u)}$, and $\boldsymbol{F^{ext}(u)}$ represent the inertial, internal, and the external force applied, respectively. The internal response to the applied external force is obtained from the stress via Equation (3).

$$\boldsymbol{F^{int}(u)} = \int \boldsymbol{B^T \sigma}\, dv \tag{3}$$

As mentioned earlier, the stress at each quadrature point is calculated by the `getStress` method of the elasto-viscoplastic material model. This method determines the stress through elasto-viscoplastic constitutive relation—relating strain to stress—using the radial return iterative technique, which iteratively computes the stress and viscoplastic strain from an elastic trial stress. Although the radial-return method is a multistep algorithm, its essence can be captured via the following:

$$\sigma_e = \sigma_e^{tr} - 3G\Delta p \tag{4}$$

Here, the actual effective stress, σ_e, is obtained by subtracting the plastic corrector, $3G\Delta p$, from the elastic predictor (i.e., effective trial stress), σ_e^{tr}. The increment of viscoplastic strain, Δp, is calculated using the following linearized viscoplastic constitutive relation:

$$\Delta p = \phi(\sigma_e, T, \epsilon_p, \rho_c, \rho_w, \Phi)\, \Delta t \tag{5}$$

where ϕ is the MoE elasto-viscoplastic ML model that depends on σ_e and T, as well as the total effective plastic strain, ϵ_p, the dislocation density within the cell, ρ_c, the dislocation density at the cell wall, ρ_w and flux, Φ. This model computes the elasto-viscoplastic strain rate, $\dot{\epsilon_p}$, and the rates of change of the cell dislocation density, $\dot{\rho_c}$ and wall dislocation density, $\dot{\rho_w}$, as shown below:

$$\textbf{outputs} = [\dot{\epsilon_p}, \dot{\rho_c}, \dot{\rho_w}] \tag{6}$$

The pseudocode presented in Algorithm 2 outlines the steps that define our "MOEElasticoViscoPlastic" material model, which implicitly integrates the elasto-viscoplastic model using return mapping.

Algorithm 2 MoEElastoViscoPlastic Material Model

```
Initialize material properties and history variables
Initialize tangent stiffness matrix from elastic properties
Instantiate MoE model wrapper
Commit initial history
procedure GETSTRESS(kinematics)
    Retrieve strain increment from kinematics
    Get history parameters and material variables
    Calculate trial stress
    Compute von Mises stress
    if von Mises stress satisfies yield criterion (non-zero for viscoplasticity) then
        Return mapping starts → to bring back the trial stress to yield surface
        Prepare input for MoE model
        Evaluate MoE model and compute Jacobian
        Perform Newton-Raphson iterations to satisfy linearized elasto-viscoplastic constitutive relation
        Update plastic strain, elastic strain, and stress
        Update history variables
        Return mapping ends
    else
        Use trial stress as final stress
    end if
end procedure
Update output parameters
Return stress, tangent stiffness matrix
```

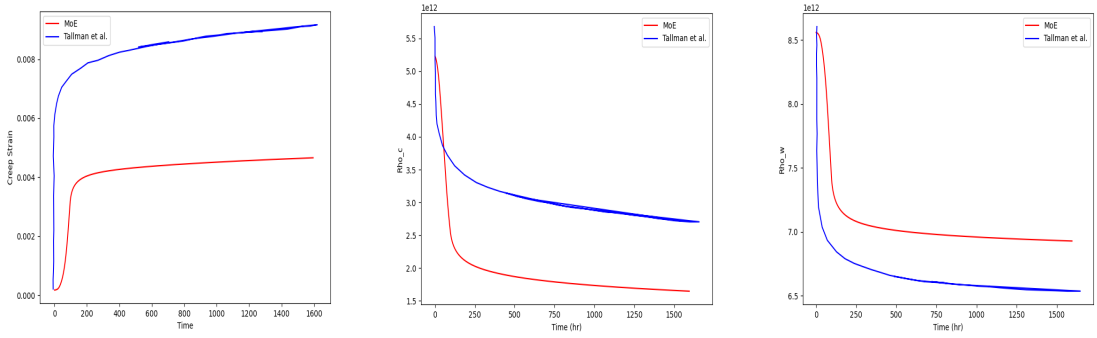
2.2. Numerical Verification

To verify the aforementioned integration of the MoE elasto-viscoplastic model within the PyFEM framework, a number of initial conditions were selected to simulate the material response using a single 3D hexahedral element.

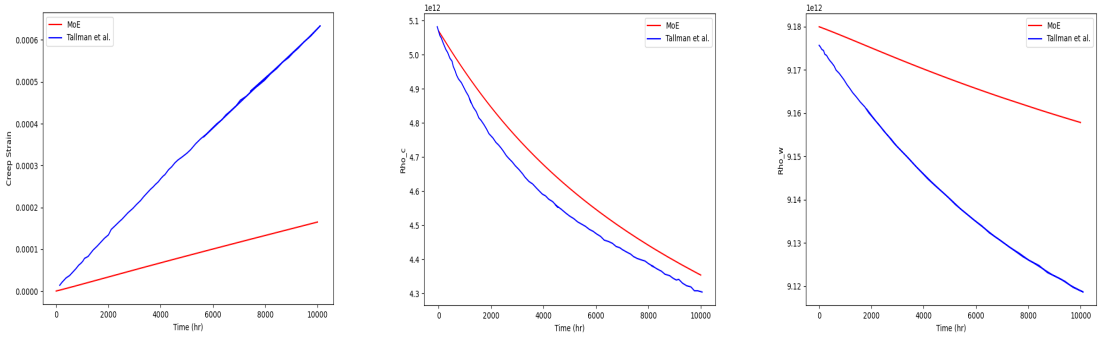
Cases 1 and 2 were taken from Tallman et al. [2], and are used here to compare results from the MoE model with results from the SM proposed before by Tallman et al. Although these are both models of the same material, they use different model forms and are based on different training data. It was found that the PyFEM MoE implementation underestimates the total strain in both the elastic and elasto-viscoplastic cases as compared to the Tallman et al. results. However, the trends shown by both of these models are similar. Their results are compared in Figure 1.

Table 1. Initial conditions for various test cases.

Case #	$\sigma_e(MPa)$	$T(K)$	$\epsilon_p(-)$	$\rho_c(m^{-2})$	$\rho_w(m^{-2})$	$\Phi(dpas^{-1})$
1	128.7	789	0.0	5.85×10^{12}	8.66×10^{12}	3.82×10^{-8}
2	50.4	607	0.0	5.07×10^{12}	9.18×10^{12}	4.83×10^{-7}
3	237.079	948.95	0.0	2.23×10^{12}	1.17×10^{13}	5.03×10^{-7}
4	169.92	982.07	0.0	5.03×10^{12}	8.45×10^{13}	8.72×10^{-7}



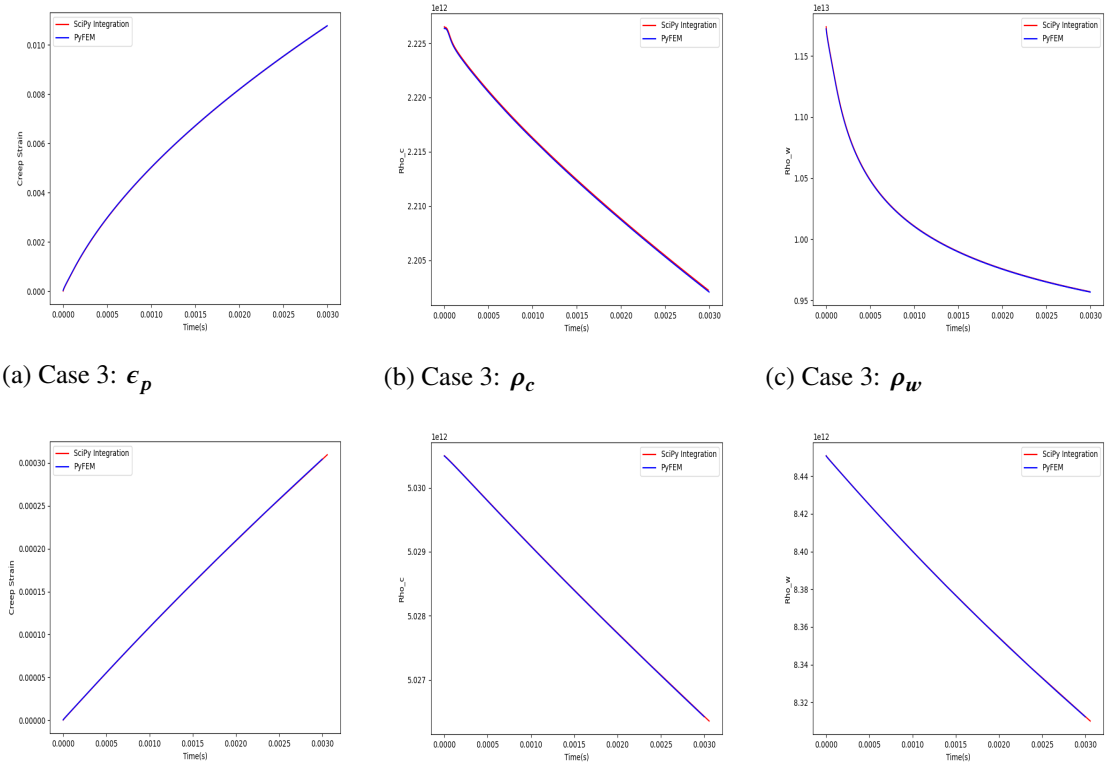
(a) Case 1: ϵ_p (b) Case 1: ρ_c (c) Case 1: ρ_w



(d) Case 2: ϵ_p (e) Case 2: ρ_c (f) Case 2: ρ_w

Figure 1. Comparison of the MoE and Tallman et al. responses for cases 1 and 2.

The implementation of the ML model within PyFEM was further assessed by comparing it with the pointwise model response, which was calculated by more directly time-integrating the MoE model using the `scipy.integrate.solve_ivp` [3] function from the SciPy library. Cases 3 and 4, taken from the viscoplastic self-consistent simulation data for model validation, were used for this purpose. The results matched with each other perfectly and thereby verified the successful integration of the elasto-viscoplastic ML model with PyFEM. Figure 2 shows the comparison between the PyFEM and SciPy integration of the MoE model.



(a) Case 3: ϵ_p (b) Case 3: ρ_c (c) Case 3: ρ_w

(d) Case 4: ϵ_p (e) Case 4: ρ_c (f) Case 4: ρ_w

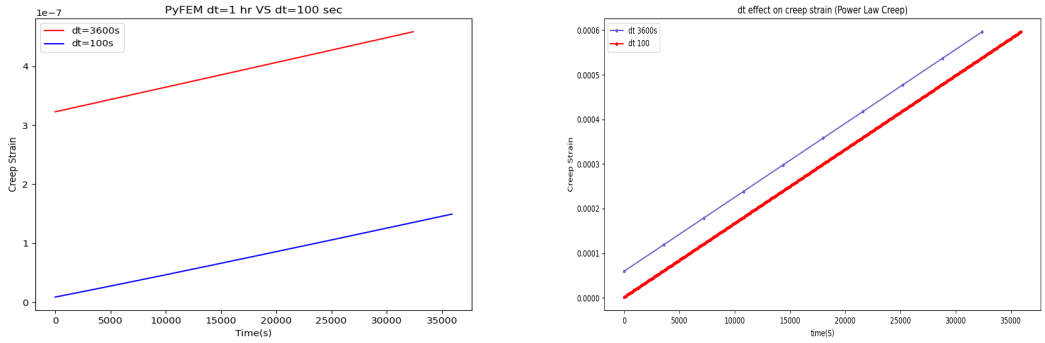
Figure 2. Comparison of the SciPy integration of the MoE model and PyFEM response for cases 3 and 4.

2.3. Investigations

ML models, especially neural networks, lack interpretability and in most cases end up functioning as a black box. Therefore, to better understand the model response in an FE setup several investigations were performed. This section describes the results and findings from these investigations.

2.3.1. Effect of Changing Time-Step Size Δt

A large set of simulations were performed using the same initial conditions, but with varying time-step sizes Δt up to 10 hours. A difference in response was observed initially between the smallest and largest time-step size for the MoE model. A reference model using the classical power-law creep model was also run, and in both the MoE and power-law creep models, it was found that there are time discretization errors when large time steps are taken, which can be attributed to the linearized representation of the viscoplastic constitutive equation in the return-mapping algorithm. However, the time discretization errors observed were insignificant in this case. The results are shown in Figure 3.



(a) MoE model

(b) Power-law creep model

Figure 3. Effects of time-step size variation. Although the gap appears large for the MoE model, the difference is negligible compared to the power-law creep model (note that the y-axis scales are different).

2.3.2. Model Surfaces

When the ML models were run, certain combinations of the initial conditions required a very small time-step size to be used to obtain a converged solution up to the desired time due to the increasing nonlinearity of the problem. Furthermore, the cases with high-stress conditions could only be run using ramped loading to avoid convergence issues. To better understand these phenomena, we plotted the surface created by the model derivative using the data obtained from the real simulations. We used case 3 and the simulation data of the dogbone sample (described in Section 2.4.1) to create the model derivative surface. We found that as the initial condition changes, so does the nonlinearity of the problem along with the smoothness of the derivative surfaces, as expected. The surfaces were created by plotting the model Jacobian (i.e., the derivative of the effective creep strain rate with respect to the von Mises stress in the creep model) against the effective stress and viscoplastic/creep strain, as shown in Figure 4 for the cases mentioned.

The model surface was also investigated. It is impossible to visualize the model surface completely because it is a multivariate function that takes six inputs and produces three outputs. However, an attempt was made to partially visualize the surface by plotting each output against the two most important inputs, namely the von Mises stress and total strain. The model inputs were varied over the range of each input as shown below:

$$\sigma_e(MPa) = [0.119132960099, 299.985430264]$$

$$T(K) = [600.0690046472204, 1099.9901302093344]$$

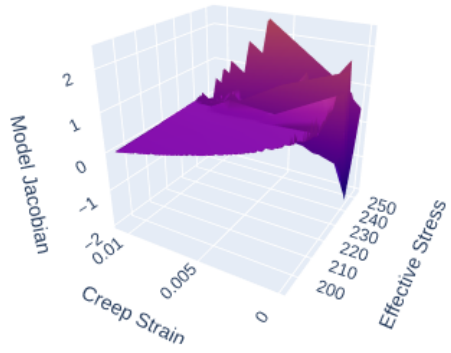
$$\epsilon_p(-) = [0.0, 0.0499999986721]$$

$$\rho_c(1/m^2) = [4669.9511535360325, 8461801410123.313]$$

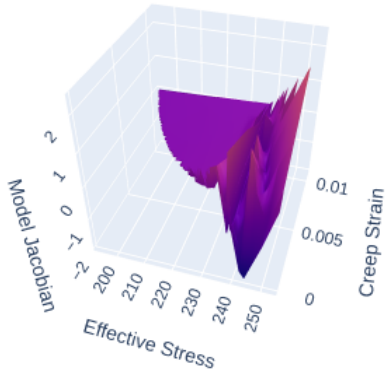
$$\rho_w(1/m^2) = [4406641771830.477, 11999567054170.322]$$

$$\Phi(dpa/s) = [1.018454806316354 \times 10^{-09}, 9.998075715064236 \times 10^{-07}]$$

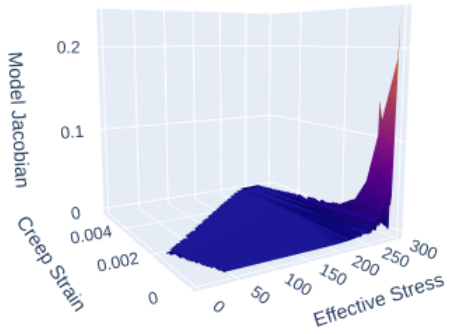
As can be seen in Figure 5, each output varies smoothly with respect to the model inputs. As expected, there is asymptotic behavior for the outputs in the vicinity of high-stress regions.



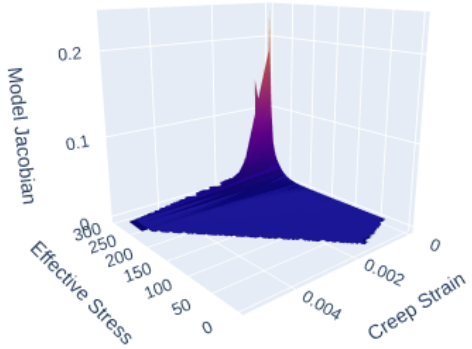
(a) Case 3: view 1



(b) Case 3: view 2

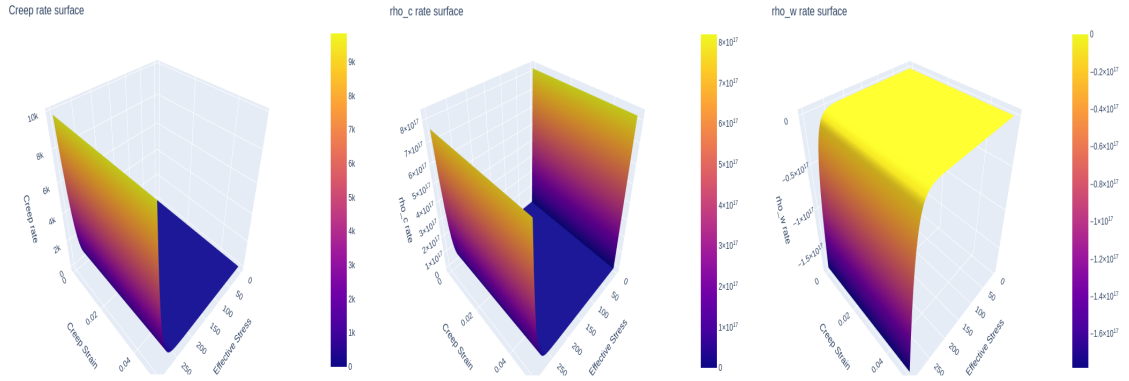


(c) Dogbone: view 1



(d) Dogbone: view 2

Figure 4. Model derivative surface obtained from PyFEM simulations.



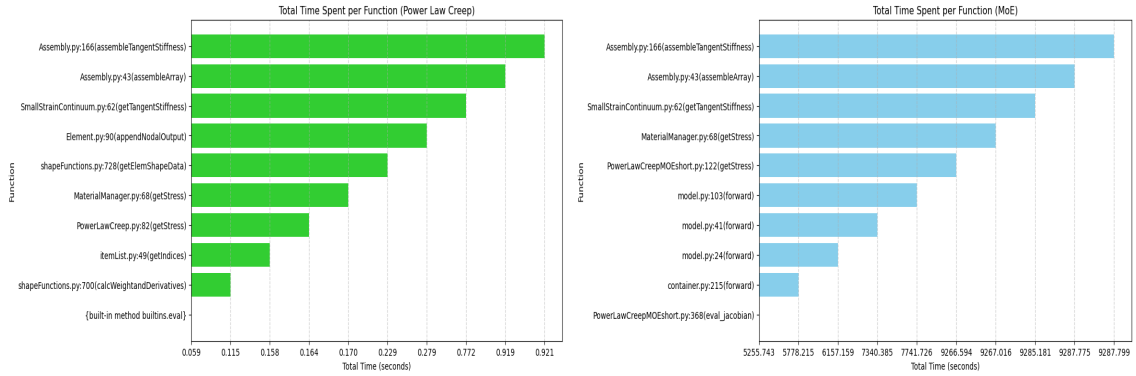
(a) $\dot{\epsilon_p}$ surface (b) $\dot{\rho_c}$ surface (c) $\dot{\rho_w}$ surface

Figure 5. Model output surfaces against effective stress and accumulated strain.

2.3.3. Time and Memory Profiling

The MoE elasto-viscoplastic material model was compared with the well-known power-law creep model as a reference case to gauge its performance in terms of timing and memory. Case 1 was simulated using both material models, ramping the stress up to 128.7 MPa and then keeping it constant. A total of 186 load steps

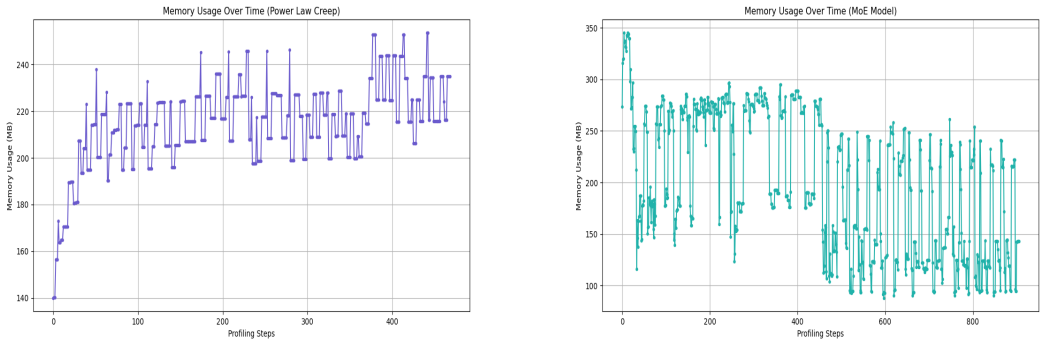
were run using a time-step size of 1 second. Although the nonlinearity of the power law was only quadratic in stress, it was found that the MoE model is slower than the power-law creep model by several orders of magnitude. While `PowerLawCreep.py` requires 0.164 seconds to run, `PowerLawCreepMoeshort.py` (the MoE model) takes a significantly longer 9,267 seconds to finish. It can be assumed with a fair amount of certainty that the internal computations of the weights and biases of the large ML model consume a significant amount of time. However, further investigations are required to identify the portions of the model that require the most computational resources. Results of this timing study are presented in Figure 6.



(a) Power-law creep model (b) MoE model

Figure 6. Timing comparison of power-law creep and ML model integrated code.

In terms of memory consumption, both models behave similarly in an average sense. Although the ML-based material model consumes comparatively more at the beginning due to MoE model instantiating, there are no obvious needs to optimize regarding memory usage at this point. Figure 7 illustrates and compares the memory consumption of both models.



(a) Power-law creep model (b) MoE model

Figure 7. Memory-usage comparison of power-law creep and ML model integrated code.

2.4. Simulations

2.4.1. Dogbone Creep

To further determine the capabilities and limitations of our current implementation, a few simulations were performed. To simulate a multiple-element entity rather than a single 3D brick element, a 16-element dogbone sample measuring $8 \times 0.5 \times 0.2$ mm was simulated under the following conditions:

$$\sigma_e = 185 \text{ MPa}$$

$$T = 789 \text{ K}$$

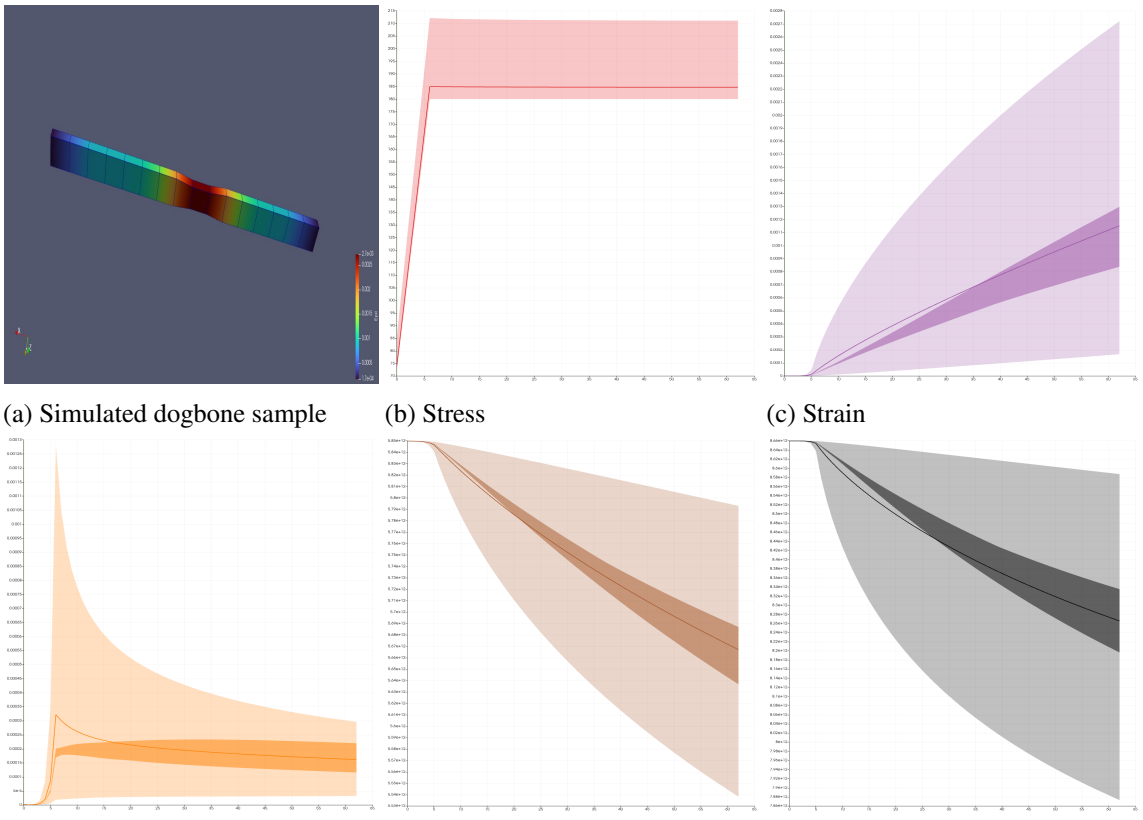
$$\epsilon_p = 0.0$$

$$\rho_c = 5.85 \times 10^{12} \text{ m}^{-2}$$

$$\rho_w = 8.66 \times 10^{12} \text{ m}^{-2}$$

$$\Phi = 3.82 \times 10^{-8} \text{ dpa s}^{-1}$$

The sample consisted of a cross-sectional area of 0.1 mm^2 at the wide region and 0.08 mm^2 at the narrow region. A tensile load of 18.5 N was applied at one end and was ramped up in six steps and held constant for 56 more steps. The results obtained from the simulation were generally acceptable. They are shown in Figure 8.



(a) Simulated dogbone sample (b) Stress (c) Strain

(d) $\dot{\epsilon_p}$ (e) ρ_c (f) ρ_w

Figure 8. (a) Simulated dogbone sample and (b-f) time histories of creep simulation outputs.

2.4.2. Stress Cycling

A stress-cycling simulation was also performed to assess the physical relevance of the integrated model under cyclic loading. Please note that the results of this simulation are preliminary, and this can be improved in the future by running up to a more physically meaningful duration where the effects of stress cycling can be assessed. We used the following conditions for our simulation with a single hexahedral element.

$$\text{Cycled between } \sigma_e = 5.184 - 128.7 \text{ MPa}$$

$$\text{Cycle time} = 200 \text{ s}$$

$$T = 789 \text{ K}$$

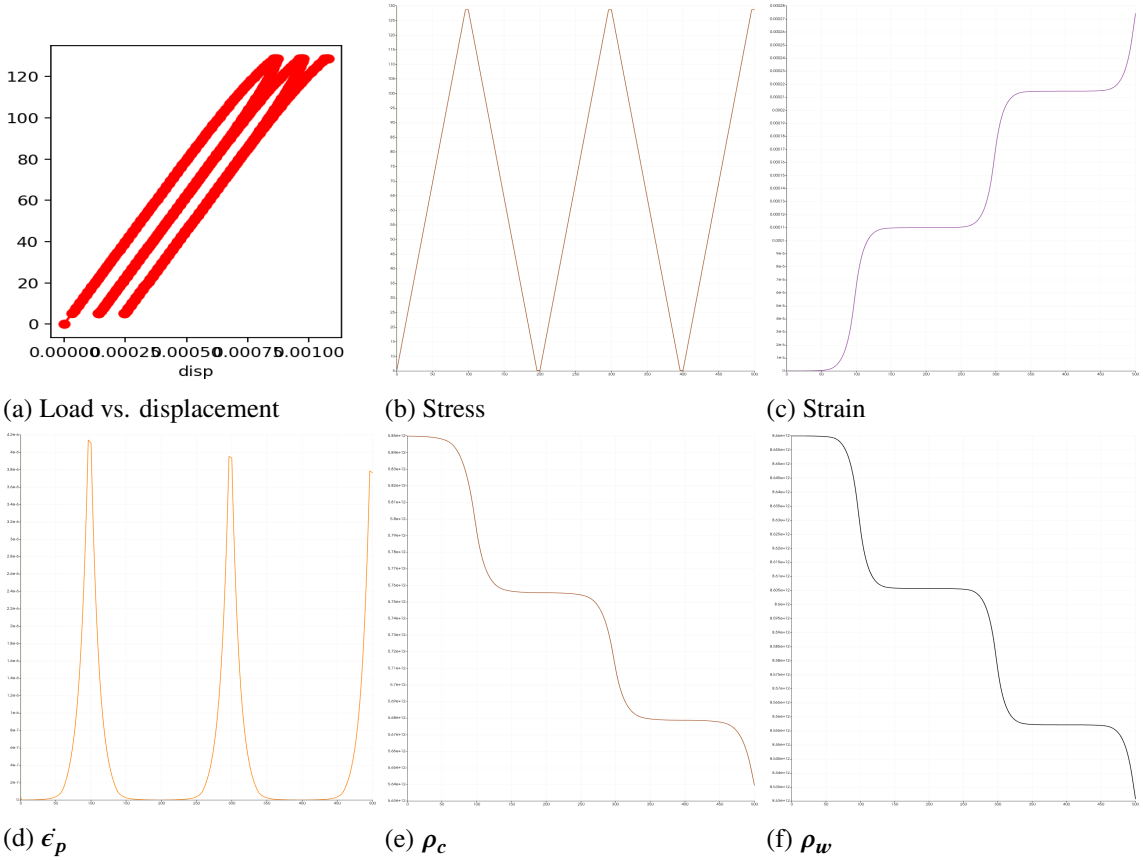
$$\epsilon_p = 0.0$$

$$\rho_c = 5.85 \times 10^{12} \text{ m}^{-2}$$

$$\rho_w = 8.66 \times 10^{12} \text{ m}^{-2}$$

$$\Phi = 3.82 \times 10^{-8} \text{ dpa s}^{-1}$$

The results from these simulations agreed well with our expectations, and no unphysical results were observed. Figure 9 illustrates the outputs from the stress-cycling simulation.



(a) Load vs. displacement (b) Stress (c) Strain

(d) $\dot{\epsilon_p}$ (e) ρ_c (f) ρ_w

Figure 9. (a) Load-displacement curve and (b-f) time histories of cyclic simulation outputs for one element.

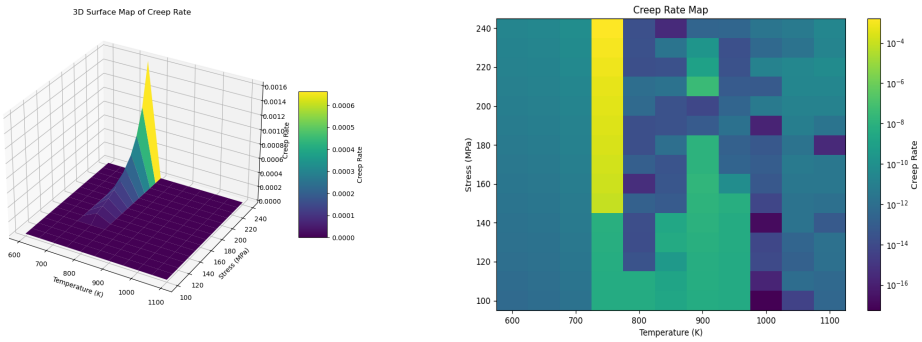
2.5. Strain Rate Maps

Strain rate maps were generated by varying the stress from 100 MPa to 240 MPa with an increase of 10 MPa, and temperatures from 600 K to 1100 K with an increase of 50 K. Therefore, a total of 165 (11×15) simulations were executed to obtain the elasto-viscoplastic strain rate for each case in the secondary stage (constant rate). For consistency, all stress cases were ramped up in 96 steps and then held constant for 1,500 steps. Figure 10 shows the time-step size of all 165 simulations. 147 of the 165 cases were simulated for up to 44.33 hours using a time-step size of 100 seconds; these cases are marked in green. A converged solution for the remaining 18 cases could only be obtained by using the time-step size shown in Figure 10, or an even smaller one.

	600	650	700	750	800	850	900	950	1000	1050	1100
100											
110											
120											
130											
140											
150				0.01							
160				0.01							
170				0.01			50				
180				0.01			50				
190				0.01			10				
200				0.01			10				
210				0.01			50				
220				0.01			50				
230				0.01			50				
240				0.01			10				

Figure 10. Time-step sizes of 165 cases. $\Delta t = 100$ s was used for all cases marked in green.

Due to the variation of Δt the resulting maps were inconsistent. The inconsistency is evident from the maps' discontinuous nature (see Figure 11).



(a) 3D surface map (b) 2D surface map

Figure 11. Generated strain rate maps.

Furthermore, depending on the combination of the initial conditions, quite a few anomalous cases were

observed where the viscoplastic strains suddenly jumped or dislocation densities dropped, or they depicted nonphysical behavior. SciPy integration of one of these cases is presented in Figure 12.

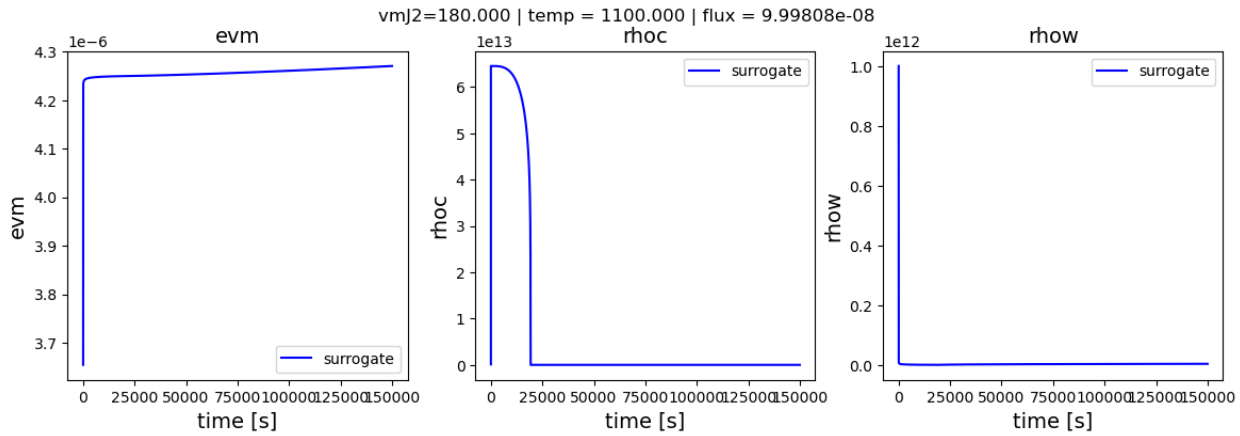


Figure 12. SciPy-integration outputs of the model over time for 180 MPa and 1100 K.

2.6. Summary and Future Work

Integrating the capabilities of ML-based constitutive models into FEM-based codes has the potential to greatly improve their ability to predict component-level response. Moreover, by using microstructural data during the model training, this approach can be used in a scale-bridging manner.

However, at this stage, a few limitations regarding the model prediction and integration still prevail. Unphysical behavior for some cases and the total simulation time required are at the forefront of them. The inconsistency and discontinuous nature of the generated strain rate maps are also of concern. Therefore future efforts will be focused on regenerating a continuous rate map and understanding the convergence issues arising from the model stiffness for some cases. In addition, in the near future, integrating the MoE elasto-viscoplastic model with the MOOSE framework [4] will be undertaken.

3. CREEP CRACK GROWTH MODELING

The primary objective of this part of the effort is to demonstrate the potential for using surrogate creep models together with XFEM to predict CCG in high-temperature alloys. The surrogate creep models are a promising method for accurately representing the deformation response in the vicinity of a crack in the presence of a variety of microstructure and macroscopic conditions. Using these creep models in conjunction with techniques such as XFEM to capture deformation ahead of a crack tip has the potential to improve CCG simulation accuracy.

3.1. Overview of Simulation Methods

This work discusses the deformation response that was modeled using a surrogate creep model, with cracks represented using the XFEM and C^* integrals used as an indicator of crack growth rate. This work was done in the MOOSE framework. The following sections summarize the key components of this simulation capability.

3.1.1. Creep Crack Growth Simulation Methods

Because of the significant plastic deformation that occurs in the region ahead of a crack tip, crack growth rates in alloys under high temperature are not well correlated with the stress intensity factor, K. The C^* integral proposed by Landes and Begley [5] was shown to be an improved indicator of CCG rates [5, 6] relative to K [7]. The C^* integral is closely related to the J integral of Rice [8], being a rate form of that integral. The time-dependent integral $C(t)$ is expressed as:

$$C(t) = \int_{\Gamma} \dot{W} \cdot \mathrm{d}x_2 - T_i \frac{\delta \dot{u}_i}{\delta x_i} \cdot \mathrm{d}s \tag{7}$$

where Γ is the integration contour, $\dot{W}$ is the strain energy density rate, x_1 and x_2 are the local coordinate directions at the crack tip (with x_1 being the direction of crack advancement), T_i is the surface traction on the crack face, u_i is the displacement vector, and s is the distance along the contour. $\dot{W}$ is defined as:

$$\dot{W} = \int_0^{\dot{\epsilon}_{mn}} \sigma_{ij} \mathrm{d}\dot{\epsilon}_{ij} \tag{8}$$

where σ_{ij} and ϵ_{ij} are the local stress and strain rates, respectively. It is important to note that while closed-form solutions for computing $\dot{W}$ for power-law creep models exist, computing this quantity is not necessarily straightforward for the data-driven types of models used here. An upper-bound approximation of $\dot{W}$ was used in the simulations presented here, but this will be revised for more accurate computations in the future.

C^* is $C(t)$ as $t \to \infty$ under steady-state conditions, and it has successfully been used as an indicator for long-term crack growth rates. Multiple consensus standards employ C^* as an indicator for CCG, and it is currently being considered for inclusion as part of a methodology for in-service surveillance in non-light-water reactors in the American Society of Mechanical Engineers Boiler and Pressure Vessel Code [9].

Despite the success of C^*-based approaches for CCG prediction, the relationships between crack growth rate $\dot{a}$ and C^* are inherently only applicable to the specific conditions of the tests on which they are based. Changing the conditions experienced by the crack, in particular the constraint and boundary conditions (e.g., load control vs. displacement control), can significantly affect the crack growth rate. Continuum damage mechanics methods that explicitly capture local damage processes in the region ahead of a crack tip have

been used with some success to predict CCG rates in fracture specimens under varying conditions, and their practical application has been demonstrated for component-level simulations [9]. The continuum damage mechanics approach requires a constitutive model that captures damage behavior. While in general it would be possible to incorporate damage into the type of surrogate creep models being developed in this project, it is unlikely that it will be feasible to do so in the timeline of the current project. Because of this, as well as the widespread acceptance of C^*-integral-based approaches, the current focus is on developing modeling approaches based on C^* integrals.

3.1.2. Extended Finite Element Method

XFEM is a technique for inserting arbitrary mesh-independent discontinuities in a continuous FE solution. It was originally proposed for modeling fracture [10, 11] but has been used for modeling a variety of other discontinuities, such as material interfaces. These interfaces can be either stationary or moving, as would be the case during processes such as corrosion.

The MOOSE framework has support for XFEM through its `xfem` module. This is one of multiple "modules" that are collections of capabilities for foundational physics models that are included in the MOOSE code distribution but are not directly part of the MOOSE framework. The `xfem` module was originally developed to target nuclear fuel fracture [12], but it has been applied to a variety of fracture-mechanics and material-interface problems. While the originally proposed formulation of XFEM employs special additional degrees of freedom at the nodes of elements traversed by interfaces to capture jump conditions, the MOOSE implementation of XFEM follows the "phantom-node" method [13]. This phantom-node method employs overlapping "partial elements" that represent nonoverlapping portions of the physical material and are connected in a way that correctly preserves the nature of the discontinuity. This method also employs special procedures to properly integrate functions over the physical portion of the partial elements. Details of the MOOSE implementation of these procedures are provided in Jiang et al. [12].

The ability of XFEM to model propagating discrete cracks independent of the FE mesh topology makes it an attractive method for CCG simulation. The representation of cracks as discrete, zero-thickness entities avoids many issues with mesh dependencies and the spurious retention of strength that can be encountered in continuum-based fracture methods. XFEM is also general in terms of the physics models that can be used in the continuous solution that it augments, and it can be used with arbitrary constitutive models, provided that stateful material property data is properly managed in the presence of propagating cracks.

3.1.3. Surrogate Creep Model

To model creep in a computationally efficient manner but capture the effects of microstructure, environment, and processing on material behavior, surrogate creep models are being used for this work to model deformation behavior in the vicinity of the crack tip. Ultimately this project will target Alloy 617 for its SM development and CCG predictions, but because the Alloy 617 SM is under development, this work is using the HT-9 model as a temporary stand-in. This project will use Alloy 617 SMs for CCG simulation efforts as they become available. The HT-9 model is based on interpolation between tabulated values of creep rate and dislocation densities as a function of stress, temperature, effective creep strain, cell dislocation density, and wall dislocation density [14]. This version of the model is significantly more robust than prior versions of the model that were based on overlapping tiles for which output quantities were interpolated using Legendre polynomials [15].

3.2. Creep Deformation Simulations with XFEM and Surrogate Models

A major objective of the work presented here is to demonstrate the ability to use XFEM together with a surrogate creep model to capture deformation response in the vicinity of cracks under a variety of conditions. This is in preparation for a longer-term objective of this project, which is to develop CCG reduced order models (ROMs) that could facilitate lifetime assessments of in-service components with observed or postulated cracks. As previously noted, the HT-9 model was used for the present demonstration.

A rectangular body modeled in 2D under plane-strain conditions with an edge crack was used for this study. The body is 0.05 m wide and 0.2 m tall. Pressure was applied in the vertical direction to put the body in tension in that direction, and, using XFEM, cracks of varying depths were prescribed on the left side of the body at its midheight. Figure 13 shows the FE mesh used for this body, as well as a zoomed-in view of the region with the crack at the end of the baseline case to show how the model captures deformation and crack opening. In most of the domain, the elements have a size of 0.002 m, but mesh refinement is used in the vicinity of the crack, with elements in that region having a size of 0.0005 m.

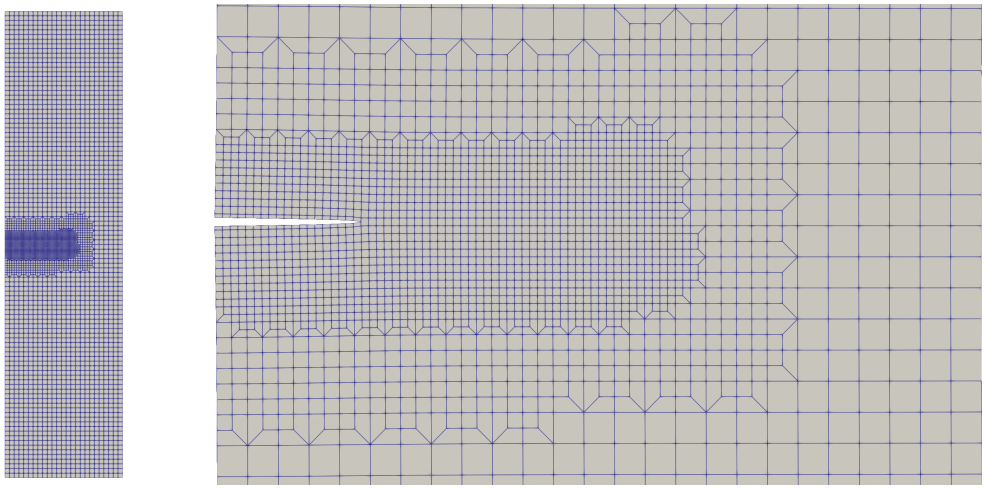


Figure 13. FE mesh of rectangular body that was modeled (left), and a zoomed-in view of the deformed mesh (right), with deformations magnified 10× in the region of the crack at the end of the baseline simulation.

The baseline simulation had a prescribed crack depth of 0.01 m, a prescribed pressure of 50 MPa at the top and bottom surfaces, a temperature of 700 K, and initial cell and wall dislocation densities of 6.0×10^{-12} m^{-2} and 1.0×10^{-13} m^{-2}, respectively. Figure 14 and Figure 15 show contours of the effective creep strain and von Mises stress at multiple times during the simulation, which was run over a time duration of approximately 6.0×10^{6} s. The von Mises stress contours show how the stress in the bulk of the material near the top and the bottom of the model is close to the applied pressure of 50 MPa, while there is a stress concentration in the vicinity of the crack. As for all fracture-mechanics problems, the peak stress is dependent on the mesh size due to the stress singularity. There are stress redistributions over the course of the simulation due to creep deformation, but these occur primarily in the vicinity of the crack tip. The creep strains, on the other hand, evolve significantly over time in both the bulk of the material and the region near the crack tip. The distinctive shape of the strain concentrations that is common in ductile fracture problems develops ahead of the crack tip.

To assess the model's ability to predict CCG under varying conditions, $C(t)$ was evaluated using this model over the course of the transient analyses under varying loading, crack depths, and microstructure parameters. To actually predict crack growth rates, experimental data on CCG is necessary. Because a placeholder material (HT-9) was used for this study, it would not be relevant to use a relationship between the

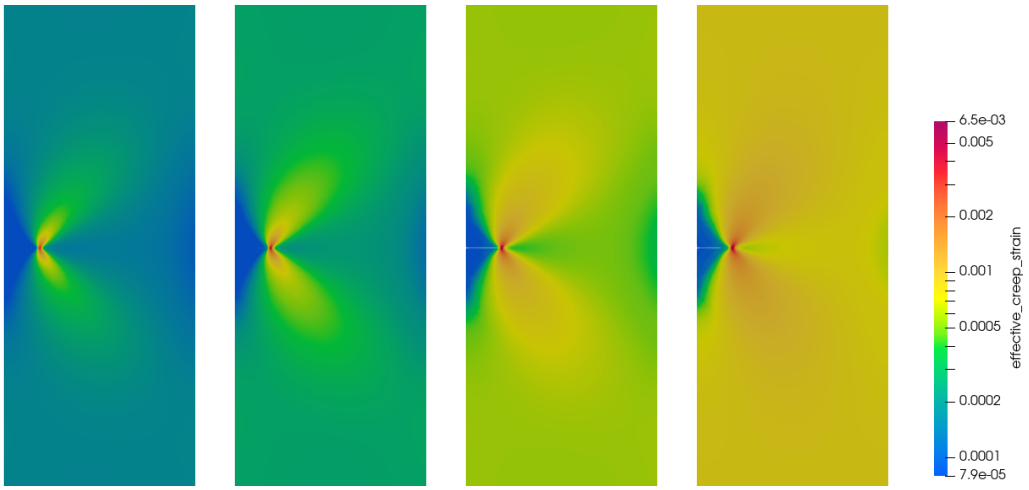


Figure 14. Contours of effective creep strain for the baseline simulation (pressure = 50 MPa, crack depth = 0.01 m, $\rho_c = 6.0 \times 10^{-12}$ m^{-2} $\rho_w = 1.0 \times 10^{-13}$ m^{-2}) at times of (from left to right) 5.0×10^5 s, 1.0×10^6 s, 3.0×10^6 s, and 6.0×10^6 s.

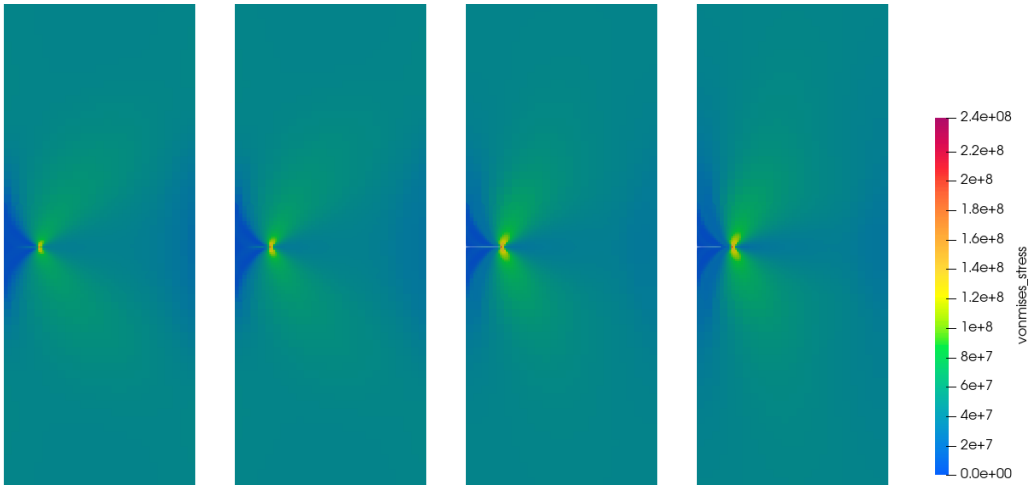


Figure 15. Contours of von Mises stress (in Pa) for the baseline simulation (pressure = 50 MPa, crack depth = 0.01 m, $\rho_c = 6.0 \times 10^{-12}$ m^{-2} $\rho_w = 1.0 \times 10^{-13}$ m^{-2}) at times of (from left to right) 5.0×10^5 s, 1.0×10^6 s, 3.0×10^6 s, and 6.0×10^6 s.

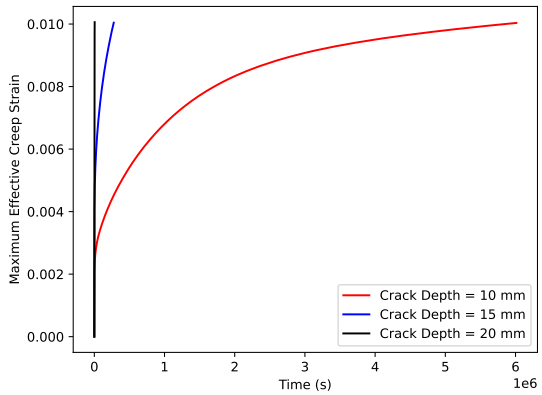
crack growth rate and C^* for another material. When a model for Alloy 617 is used with this model later in this project, an effort will be made to use all available CCG data for that alloy. There are ongoing efforts in the U.S. Department of Energy's Advanced Reactor Technologies program to collect data on CCG in Alloy 617 [16, 17], and that data will be used to the extent it is applicable.

To study the model's sensitivity to crack depth, crack depths of 0.01, 0.015, and 0.02 m were simulated using the baseline set of parameters. To study its sensitivity to applied pressure, pressures of 10, 30, and 50 MPa were applied. Finally, to assess the effects of microstructure parameters, three different sets of cell and wall dislocation densities (ρ_c and ρ_w, respectively) were used: a "low" case with values of 4×10^{12} m^{-2} and 8×10^{12} m^{-2}, respectively, a "medium" case with values of 5×10^{12} m^{-2} and 9×10^{12} m^{-2}, respectively, and a "high" case with values of 6×10^{12} m^{-2} and 1×10^{13}, respectively. Figure 16 shows time histories of the maximum effective creep strain and $C(t)$ for these studies.

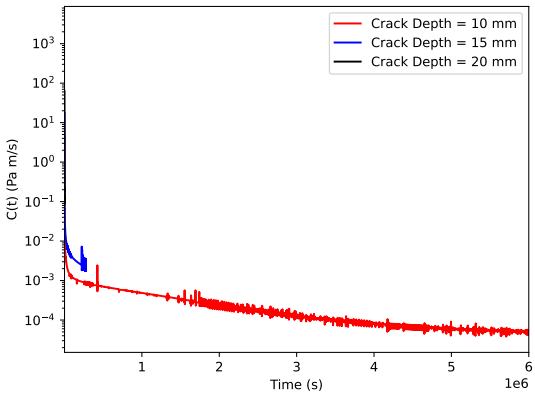
It is important to note that the measure of effective creep strain is mesh dependent, but it is still useful for comparing the near-tip response of the material in these cases. Also, as previously noted, an upper-bound estimate was used to calculate the strain energy rate density in $C(t)$, and further development is needed to more accurately calculate this for these SMs. As would be intuitively expected, larger crack depths lead to higher stress intensity under the same far-field load, along with higher effective creep strain. The model terminates when a limiting value of 0.01 is reached for the effective creep strain, and that value is reached much earlier for the deeper cracks. The $C(t)$ integral decreases over time, and the deeper crack has higher values of $C(t)$, at least for the time range studied.

Trends similar to those observed for varying crack depth were observed for increasing pressure. As would be expected, the maximum effective creep strain increases with increasing pressure, which corresponds to an increasing far-field stress, and at least initially C(t) is also higher with increasing pressure. Interestingly, $C(t)$ also follows that same trend initially, with higher values for higher applied pressure. However, later in time the value for 50 MPa drops below that for 30 MPa. This needs further investigation, but potentially it could be due to the creep response going into a different regime as the material's internal state evolves.

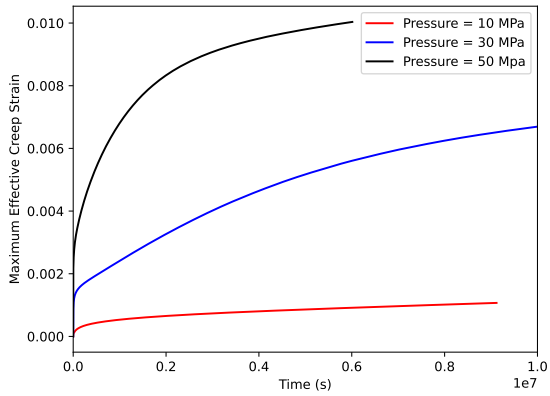
The study of microstructure dependence, where the dislocation densities were changed, shows that with increasing dislocation density the effective creep strain increases, as does $C(t)$, which is indicative of increased deformation near the crack tip. A crossover behavior for the highest dislocation content (which is the baseline case in the other simulations) similar to that observed in the pressure-dependent response was observed in these simulations, and could be occurring due to similar reasons.



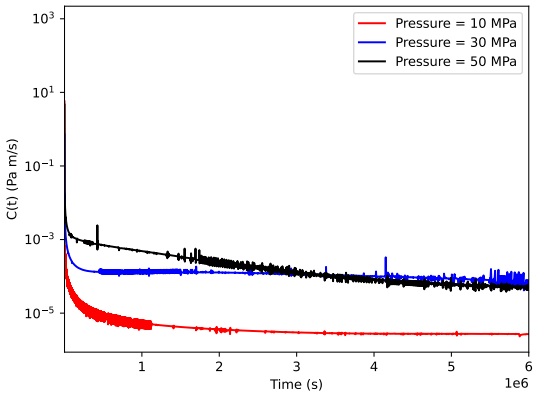
(a) Crack depth: maximum effective creep strain



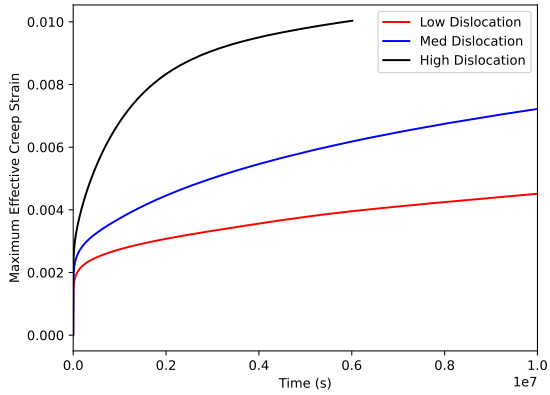
(b) Crack depth: $C(t)$



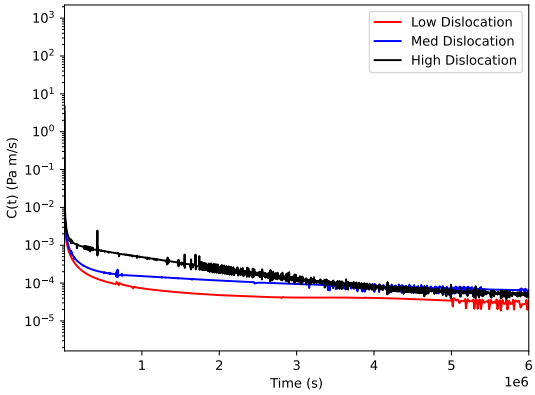
(c) Pressure: maximum effective creep strain



(d) Pressure: $C(t)$



(e) Microstructure: maximum effective creep strain



(f) Microstructure: $C(t)$

Figure 16. Time histories of maximum effective creep strain and $C(t)$ for the parameter studies for crack depth, applied pressure, and microstructure (parameterized by dislocation density).

3.3. Summary and Future Work

This effort demonstrated using XFEM with the SM to predict the elastic-viscoplastic response of HT-9 steel in the vicinity of a crack. This study demonstrated that these modeling approaches can be used together, and that this modeling approach is sufficiently robust for exploring the effects of modeling parameters. The effects of varying crack depth, applied pressure, and microstructure parameters were studied, and for the most part the results were as expected, although there was a somewhat unexpected crossover between the computed values of $C(t)$ for some of the cases, which warrants further exploration.

Logical next steps for this effort include refining the procedure for computing $C(t)$ for SMs, utilizing the Alloy 617 SM when it becomes available, and locating relevant CCG data for Alloy 617 and using it for predicting crack growth rates. In addition, further testing of this modeling approach is needed to ensure its validity. This could be done using data for Alloy 316H, for which there is experimental CCG data available. Longer-term objectives could include using a continuum damage mechanics approach for CCG with the surrogate creep model.

4. REFERENCES

[1] F. P. Dunne and N. Petrinić, *Introduction to computational plasticity*. 6 2005.

[2] A. E. Tallman, M. A. Kumar, C. Matthews, and L. Capolungo, "Surrogate modeling of viscoplasticity in steels: application to thermal, irradiation creep and transient loading in HT-9 cladding," *JOM*, vol. 73, pp. 126–137, 10 2020.

[3] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors, "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python," *Nature Methods*, vol. 17, pp. 261–272, 2020.

[4] G. Giudicelli, A. Lindsay, L. Harbour, C. Icenhour, M. Li, J. E. Hansel, P. German, P. Behne, O. Marin, R. H. Stogner, J. M. Miller, D. Schwen, Y. Wang, L. Munday, S. Schunert, B. W. Spencer, D. Yushu, A. Recuero, Z. M. Prince, M. Nezdyur, T. Hu, Y. Miao, Y. S. Jung, C. Matthews, A. Novak, B. Langley, T. Truster, N. Nobre, B. Alger, D. Andrs, F. Kong, R. Carlsen, A. E. Slaughter, J. W. Peterson, D. Gaston, and C. Permann, "3.0 - MOOSE: Enabling massively parallel multiphysics simulations," *SoftwareX*, vol. 26, p. 101690, May 2024.

[5] J. D. Landes and J. A. Begley, *A Fracture Mechanics Approach to Creep Crack Growth*, p. 128–148. ASTM International, Jan. 1976.

[6] K. M. Nikbin, G. A. Webster, and C. E. Turner, *Relevance of Nonlinear Fracture Mechanics to Creep Cracking*, p. 47–62. ASTM International, Jan. 1976.

[7] M. J. Siverns and A. T. Price, "Crack propagation under creep conditions in a quenched 2 1/4 chromium 1 molybdenum steel," *International Journal of Fracture*, vol. 9, pp. 199–207, June 1973.

[8] J. R. Rice, "A path independent integral and the approximate analysis of strain concentration by notches and cracks," *Journal of Applied Mechanics*, vol. 35, no. 2, p. 379, 1968.

[9] A. Recuero, M. Petkov, B. W. Spencer, and P.-A. Juan, "Continuum damage mechanics modeling of high-temperature flaw propagation: Application to creep crack growth in 316H standardized specimens and nuclear reactor components," *Journal of Pressure Vessel Technology*, pp. 1–26, July 2023.

[10] T. Belytschko and T. Black, "Elastic crack growth in finite elements with minimal remeshing," *International Journal for Numerical Methods in Engineering*, vol. 45, pp. 601–620, June 1999.

[11] N. Moës, J. Dolbow, and T. Belytschko, "A finite element method for crack growth without remeshing," *International Journal for Numerical Methods in Engineering*, vol. 46, pp. 131–150, Sept. 1999.

[12] W. Jiang, B. W. Spencer, and J. E. Dolbow, "Ceramic nuclear fuel fracture modeling with the extended finite element method," *Engineering Fracture Mechanics*, vol. 223, p. 106713, Jan. 2020.

[13] A. Hansbo and P. Hansbo, "A finite element method for the simulation of strong and weak discontinuities in solid mechanics," *Computer Methods in Applied Mechanics and Engineering*, vol. 193, pp. 3523–3540, Aug 2004.

[14] A. Ruybalid, R. Sweet, L. Capolungo, and B. Eftink, "Development and integrate into MOOSE a HT-9 constitutive surrogate creep model," Tech. Rep. TODO, Los Alamos National Laboratory, Aug. 2024.

[15] R. Sweet and S. Novascone, "Evaluation of reduced order model for HT-9 creep and modifications to current HT-9 creep model in BISON," Tech. Rep. INL/RPT-22-70168, Idaho National Laboratory, Sept. 2022.

[16] M. McMurtrey, M. Heighes, and C. Gibson, "Results of FY 2023 Alloy 617 and Alloy 709 high-temperature crack-growth testing," Tech. Rep. INL/RPT-23-74791, Idaho National Laboratory, Sept. 2023.

[17] M. McMurtrey and M. Heighes, "Summary of results from Alloy 617 and Alloy 709 elevated-temperature crack-growth-test from planned FY-24 activities," Tech. Rep. INL/RPT-24-81003, Idaho National Laboratory, Sept. 2024.