DISCLAIMER

Surrogate Model Integration with MOOSE XFEM for Creep Crack Growth

INL/RPT-25-89212

Scientific Discovery through Advanced Computation

SEPTEMBER 2025

K M Zaheen Nasir and Wen Jiang

North Carolina State University

Giang D. Huynh and 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-25-89212

Surrogate Model Integration with MOOSE XFEM for Creep Crack Growth

K M Zaheen Nasir and Wen Jiang

North Carolina State University

Giang D. Huynh and Benjamin W. Spencer

Idaho National Laboratory

September 2025

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

Ferritic-martensitic steels are key structural materials for advanced reactors but experience time-dependent deformation and damage under prolonged high temperature and irradiation, leading to creep-driven crack initiation and growth. High-fidelity models—crystal plasticity with irradiation mechanisms, phase-field for microstructural evolution, and continuum-damage viscoplasticity—capture the underlying physics but are too computationally intensive for broad design-space exploration and uncertainty quantification. This milestone advances a scalable alternative by integrating a microstructure-sensitive surrogate creep model into the Multiphysics Object-Oriented Simulation Environment (MOOSE) finite element framework and extending it to fracture via the extended finite element method (XFEM). The surrogate model, developed with collaborators at Sandia and Los Alamos National Laboratories, maps relevant microstructural descriptors to the viscoplastic response of HT9. We embed this surrogate within a coupled deformation-damage workflow in MOOSE/XFEM to simulate creep-driven crack initiation and propagation. Implementation enhancements include updates to the material interface, a plastic correction phase involving microstructure evolution, and fracture criteria to ensure numerical robustness and compatibility with the surrogate structure. Demonstrations on canonical creep benchmarks spanning uniaxial and multiaxial states show that the surrogate reproduces key trends of high-fidelity models while substantially reducing computational cost. The resulting capability bridges physics fidelity and performance, providing a practical path to a predictive, microstructure-aware assessment of creep and fracture in reactor materials.

Page intentionally left blank

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 U.S. 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

ACRONYMS

AD	automatic differentiation
CCG	creep crack growth
DOF	degree of freedom
FE	finite element
ML	machine learning
MNR-RR	radial return with multivariate Newton–Raphson
MoE	Mixture of Experts
MOOSE	Multiphysics Object-Oriented Simulation Environment
NR	Newton–Raphson
QoI	quantity of interest
ROM	reduced order model
RR	radial return
RSM	response surface method
SERD	strain energy release density
UNR-RR	radial return with univariate Newton–Raphson
VPSC	visco-plastic self consistent
XFEM	extended finite element method

Page intentionally left blank

Surrogate Model Integration with MOOSE XFEM for Creep Crack Growth

1. INTRODUCTION

Alloys deployed for structural components in advanced nuclear reactors have demanding requirements due to their operating environment, which can include a combination of exposure to radiation, high temperature, and corrosive materials [1, 2]. Because of this, a limited set of alloys is currently available for high-temperature nuclear applications. Under prolonged exposure to elevated temperatures and neutron irradiation, these alloys undergo time-dependent deformation (creep). Additionally, when internal stresses exceed the material strength, creep damage can occur, in a process that involves nucleating and growing microcracks, and can eventually lead to component failure. The macroscopic response of high-temperature alloys is strongly influenced by their evolving microstructure, including dislocation networks, grain boundaries, and precipitate populations [3, 4].

Experimental programs are essential for model calibration and validation. However, the high cost, long duration, and limited flexibility of creep testing constrain coverage of the full design space [5, 6]. Consequently, modeling and simulation play a key role in accelerating materials qualification and providing predictive capabilities that can be applied to understand the performance of components under reactor conditions. This report documents progress on developing engineering-scale capabilities for modeling creep and creep crack growth in high-temperature alloys for the Simulation of the Response of Structural Metals in Molten Salt Environment project under the U.S. Department of Energy's Scientific Discovery through Advanced Computing (SciDAC) program. Because of the availability of relevant models and data, initial efforts in this project, including those documented here, have focused on the HT9 alloy, and later in this project these methods and tools will be applied to Alloy 617.

Modeling creep in HT9 across relevant length and time scales is challenging due to the coupled action of dislocation glide/climb, grain-boundary sliding, and void nucleation [7]. Mechanistic approaches have previously been developed to capture these processes with high fidelity. For example, crystal-plasticity frameworks incorporating dislocation-climb laws informed by rate theory have been used to simulate thermal and irradiation creep in HT9, including stress-induced preferential absorption and other irradiation-specific effects [8]. Phase-field models have been employed to simulate microstructural evolution under creep—such as void nucleation/growth, grain coarsening, and damage localization—enabling spatially resolved predictions of damage and crack initiation [9, 10]. Continuum damage mechanics–based viscoplastic models also describe the coupling between creep deformation and damage evolution, particularly under multiaxial and cyclic loads [11]. Despite their physical rigor, these models are computationally intensive and often impractical for large-scale simulations or design optimization over many loading scenarios.

To address these limitations and enable accurate component-scale simulations, surrogate models have emerged as a promising approach. They can emulate high-fidelity behavior at a fraction of the computational cost to enable rapid design exploration and uncertainty quantification. In this project, a microstructure-sensitive surrogate creep model has been developed by collaborators at Sandia and Los Alamos National Laboratories. This model, which is based on a Mixture of Experts (MoE) approach, was previously tested in a standalone constitutive driver code and in a Python finite element (FE) code to facilitate initial testing [12]. For practical component-scale simulations, it is important to have this model integrated into a high-performance FE code.

The effort documented here builds on the prior work through developments in two key areas. First, we

integrated the microstructure-sensitive MoE surrogate creep model into the Multiphysics Object-Oriented Simulation Environment (MOOSE) FE framework (Section 2). This model captures the viscoplastic response of HT9 as a function of microstructural descriptors. Second, we utilized this surrogate in conjunction with the extended finite element method (XFEM) in MOOSE to simulate creep-driven crack propagation (Section 3). To enable these capabilities, we implemented enhancements for compatibility with the surrogate's structure, including updates to the material interface, time-integration schemes, and crack-growth criteria. Collectively, these advances represent important steps toward enabling accurate and computationally efficient simulations of creep and fracture in reactor alloys.

2. CREEP SURROGATE MODELING

Recently, Sandia National Laboratories and Los Alamos National Laboratory collaborated to create an MoE [13] machine learning (ML) model [14], trained on data from visco-plastic self consistent (VPSC) [4, 15] mesoscale simulations, to predict the elasto-viscoplastic behavior of HT9 steel under reactor operating conditions, where each expert is a neural network. Still, this model must be seamlessly integrated into an FE framework in a manner that would enable us to leverage its superior predictive ability for simulating a realistic physical component. Regarding the integration of ML surrogates into the FE method, most existing approaches remain largely restricted to fully offline coupling (where the results obtained beforehand from the ML surrogate are fed to the FE solver). This necessitates either extra post-processing before the FE solution is carried out, employing on straightforward data sampling strategies wherein the ML constitutive model is employed to build the database, or eliminating plastic correction steps altogether. In the current literature, implicit integration schemes that incorporate ML models as constitutive relations within the return mapping framework for iterative plastic correction are rarely used. Consequently, in this effort we successfully implemented an implicit integration of the said model, adapting the method of radial return (RR) in an FE framework. In addition to our previous implementation [12] in a standalone Python-based FE code, PyFEM, the open-source MOOSE [16] was selected as a host FE framework due to its ease of scalability, modularity, and multiphysics coupling. Details of the MoE model development effort are described as follows: Section 2.1 describes its mathematical formulations and accompanying algorithms. Section 2.2 discusses the careful verification of our implementations over the range of the ML model's input space. Section 2.3.1 describes how the MoE surrogate model and its FE integration surpass its predecessor response surface method (RSM)-based surrogate. Finally, Section 2.4 provides concluding remarks and addresses limitations and future work.

2.1. Methods and Algorithms

2.1.1. Mathematical Preliminaries

We start by providing the governing equation associated with the quasistatic elasto-viscoplastic problem of our domain. This is the well-known equation of stress analysis given in Equation 1, where $\boldsymbol{b}$ denotes the body forces and $\boldsymbol{\sigma}$ denotes the Cauchy stress tensor.

$$\nabla \cdot \boldsymbol{\sigma} + \boldsymbol{b} = 0 \tag{1}$$

In its simplest form, neglecting body forces, we have the Cauchy equilibrium force balance equation expressed as [17]

$$\nabla \cdot \boldsymbol{\sigma} = \begin{bmatrix} \frac{\partial \sigma_{xx}}{\partial x} + \frac{\partial \sigma_{xy}}{\partial y} + \frac{\partial \sigma_{xz}}{\partial z} \\ \frac{\partial \sigma_{yx}}{\partial x} + \frac{\partial \sigma_{yy}}{\partial y} + \frac{\partial \sigma_{yz}}{\partial z} \\ \frac{\partial \sigma_{zx}}{\partial x} + \frac{\partial \sigma_{zy}}{\partial y} + \frac{\partial \sigma_{zz}}{\partial z} \end{bmatrix} = 0, \tag{2}$$

where $\boldsymbol{\sigma}$ obeys the constitutive relation expressed in Equation 3. Here, $\boldsymbol{C}$ is the fourth-order elasticity or stiffness tensor relating elastic strain to stress. In all succeeding simulations presented here, isotropic material response is assumed. Therefore, the fourth-order stiffness tensor $\boldsymbol{C}$ can essentially be expressed in terms of only two elastic constants [18].

$$\boldsymbol{\sigma} = \boldsymbol{C} : \boldsymbol{\varepsilon}^e \tag{3}$$

The strain tensor $\boldsymbol{\varepsilon}$ can be additively decomposed into its elastic and plastic parts:

$$\boldsymbol{\varepsilon} = \boldsymbol{\varepsilon}^e + \boldsymbol{\varepsilon}^p. \tag{4}$$

For our material of interest, plasticity includes the viscoplastic behavior as well. Therefore, all forthcoming formulations consider viscoplastic cases, where $\boldsymbol{\varepsilon}^{\boldsymbol{vp}}$ denotes the viscoplastic strain tensor. Hence, Equation 3 becomes

$$\boldsymbol{\sigma} = \boldsymbol{C} : (\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}^{vp}). \tag{5}$$

In incremental form, Equation 5 can be written for the stress tensor at the current time step $i + 1$ as follows:

$$\boldsymbol{\sigma}_{i+1} = \boldsymbol{\sigma}_i + \Delta\boldsymbol{\sigma} = \boldsymbol{\sigma}_i + \boldsymbol{C} : (\Delta\boldsymbol{\varepsilon} - \Delta\boldsymbol{\varepsilon}^{vp}). \tag{6}$$

We determine the incremental viscoplastic strain tensor, $\Delta\boldsymbol{\varepsilon}^{vp}$, from the scalar incremental viscoplastic strain, Δp, by using the Prandtl–Reuss normality rule from the flow rule below:

$$\Delta\boldsymbol{\varepsilon}^{vp} = \frac{3}{2}\,\Delta p\,\frac{\boldsymbol{S}}{\sigma_e}, \tag{7}$$

where $\boldsymbol{S}$ is the deviatoric stress tensor and σ_e denotes the effective stress. The key remaining task is then to determine the scalar incremental viscoplastic strain, Δp, by utilizing an appropriate constitutive relationship (in contrast to a consistency condition used for rate-independent plasticity) that is representative of the material at hand (in our case, HT9 steel). The viscoplastic rate, characterized by Δp, evolves for any non-zero stress. In this endeavor, we replace this constitutive relationship with an ML surrogate model, $\boldsymbol{\mathcal{M}}$, described in the following equation:

$$\boldsymbol{\mathcal{M}} : \left(\sigma_e,\, T,\, \varepsilon_{vm},\, \rho_c,\, \rho_w,\, \Phi\right) \mapsto \left(\dot{\varepsilon}_{vp},\, \dot{\rho_c}, \dot{\rho_w}\right). \tag{8}$$

This maps the state variables effective stress, σ_e; temperature, T; effective von Mises strain, ε_{vm}; dislocation density at cell, ρ_c; dislocation density at cell wall, ρ_w; and neutron flux, Φ, to the rate-related information (i.e., elasto-viscoplastic strain rate, $\dot{\varepsilon}_{vp}$, and the dislocation density change rates, $\dot{\rho_c}$ and $\dot{\rho_w}$).

2.1.2. Implicit Integration Schemes

As the yield surface is smeared into the elasto-viscoplastic constitutive surrogate model, $\boldsymbol{\mathcal{M}}$ for any non-zero stress, and normality is imposed through Equation 7, we follow the RR method [17] to implicitly integrate the elasto-viscoplastic constitutive relation. At the start of the return mapping, we assume the entire strain increment to be fully elastic. Therefore, the $(i + 1)^{th}$ current stress tensor at time step $i + 1$ is given by

$$\boldsymbol{\sigma}^{tr}_{i+1} = \boldsymbol{\sigma}^{tr}_i + \boldsymbol{C} : \Delta\boldsymbol{\varepsilon}_{total}. \tag{9}$$

Consequently, we inform the initial state variables:

$$\sigma_e = \sigma^{tr}_e; \;\; \boldsymbol{S} = \boldsymbol{S}^{tr}.$$

We also linearize the elasto-viscoplastic constitutive equation in time to obtain the scalar incremental viscoplastic strain, Δp, and update the state variables (dislocation densities in the case of the current model) for a given increment. After determining Δp using an iterative Newton–Raphson (NR) process, the actual effective stress is calculated as follows according to the return-mapping formalism [17]:

$$\sigma_e = \sigma^{tr}_e - 3G\Delta p. \tag{10}$$

The viscoplastic strain increment, Δp, is always implicitly updated using an NR iteration scheme. However, depending on the choice of updating dislocation densities (explicitly or implicitly), we proposed two formulations along with their accompanying algorithms, namely radial return with univariate Newton–Raphson (UNR-RR) (also known as scalar-valued return mapping) and radial return with multivariate Newton–Raphson (MNR-RR) (also known as general return mapping). While the univariate approach (UNR-RR) adapted to the previous RSM surrogate was previously implemented, the multivariate formulation (MNR-RR) is a novel addition proposed in this work. It is superior in terms of accuracy compared to the UNR-RR, which has been used for its predecessors [19, 20] and in previous investigations [12]. The UNR-RR and MNR-RR formulations are outlined in Algorithms 1 and 2, respectively.

Algorithm 1 UNR-RR Scheme

Input: Shear modulus μ, Temperature T, Initial dislocation densities (ρ_c, ρ_w), Neutron flux Φ
Output: Stress $\boldsymbol{\sigma}$, Visco-plastic strain ε_{vp}, Updated dislocation densities
Load ML Model from file (only once)
initQpStatefulProperties()→ Initialize $\rho_c = \rho_{c0}$, $\rho_w = \rho_{w0}$
computeQpStress():
for each quadrature point **do**
  Compute Elastic Strain Increment: $\Delta\boldsymbol{\varepsilon}_{\boldsymbol{fully\_elastic}} = \boldsymbol{\varepsilon}^{\boldsymbol{new}}_{\boldsymbol{mech}} - \boldsymbol{\varepsilon}^{\boldsymbol{old}}_{\boldsymbol{mech}}$
  Compute Trial Stress: $\boldsymbol{\sigma}_{\boldsymbol{trial}} = \boldsymbol{\sigma}_{\boldsymbol{old}} + \boldsymbol{C} : \Delta\boldsymbol{\varepsilon}_{\boldsymbol{total}}$
  Compute Effective Stress: $\sigma_e = \sqrt{\frac{3}{2}\boldsymbol{S} : \boldsymbol{S}}$
  if $\sigma_e > 1 +$ tolerance **then**
    Prepare Input for Model: $\boldsymbol{\mathcal{F}} = [\sigma_e, T, \varepsilon^{old}_{vm}, \rho^{old}_c, \rho^{old}_w, \Phi)]^T$
    Predict Creep Rate $\dot{\varepsilon}_{vp}$ using ML Model
    Perform **Return Mapping** (Newton-Raphson Iteration):
    while Residual > *tol* or first iteration **do**
      Residual: $res = \Delta p - \dot{\varepsilon}_{vp}\Delta t$
      Update: $\Delta p \leftarrow \Delta p - \frac{res}{J}$
      Recompute Visco-plastic Rate with updated input
    end while
    Update State Variables:

$$\varepsilon_{vm} = \varepsilon^{old}_{vm} + \Delta p$$

$$\rho_c = \rho^{old}_c + \dot{\rho_c}\Delta t$$

$$\rho_w = \rho^{old}_w + \dot{\rho_w}\Delta t$$

  end if
  Compute Incremental Viscoplastic Strain Tensor: $\Delta\boldsymbol{\varepsilon}^{\boldsymbol{vp}} = \frac{3}{2}\,\Delta p\,\frac{\boldsymbol{S}^{\boldsymbol{tr}}}{\sigma^{tr}_e}$
  Compute Final Stress: $\boldsymbol{\sigma} = \boldsymbol{\sigma}_{\boldsymbol{old}} + \boldsymbol{C} : (\Delta\boldsymbol{\varepsilon}_{\boldsymbol{total}} - \Delta\boldsymbol{\varepsilon}^{\boldsymbol{vp}})$
end for

Algorithm 2 MNR-RR Scheme

Input: Shear modulus μ, Temperature T, Initial dislocation densities (ρ_c, ρ_w), Neutron flux Φ
Output: Stress $\boldsymbol{\sigma}$, Visco-plastic strain ε_{vp}, Updated dislocation densities
Load ML Model from file (only once)
initQpStatefulProperties()→ Initialize $\rho_c = \rho_{c0}$, $\rho_w = \rho_{w0}$
computeQpStress():
for each quadrature point **do**
  Compute Elastic Strain Increment: $\Delta\boldsymbol{\varepsilon}_{fully\_elastic} = \boldsymbol{\varepsilon}^{new}_{mech} - \boldsymbol{\varepsilon}^{old}_{mech}$
  Compute Trial Stress: $\boldsymbol{\sigma}_{trial} = \boldsymbol{\sigma}_{old} + \boldsymbol{C} : \Delta\boldsymbol{\varepsilon}_{total}$
  Compute Effective Stress: $\sigma_e = \sqrt{\frac{3}{2}\boldsymbol{S} : \boldsymbol{S}}$
  if $\sigma_e > 1 +$ tolerance **then**
    Define vector of **QoI**: $\boldsymbol{q} = \begin{bmatrix} \Delta p \\ \Delta\rho_c \\ \Delta\rho_w \end{bmatrix}$
    Prepare Input for Model: $\boldsymbol{\mathcal{I}} = [\sigma_e, T, \varepsilon^{old}_{vm}, \rho^{old}_c, \rho^{old}_w, \Phi)]^T$
    Predict Outputs/Rate vector using ML Model: $\boldsymbol{\mathcal{O}}(\boldsymbol{q}) = \begin{bmatrix} \dot{\varepsilon}_{vp} \\ \dot{\rho}_c \\ \dot{\rho}_w \end{bmatrix} = \boldsymbol{\mathcal{M}}(\boldsymbol{\mathcal{I}})$
    Perform **Return Mapping** (Newton-Raphson Iteration):
    while Residual norm $>$ *tol* or first iteration **do**
      Residual: $\boldsymbol{r}(\boldsymbol{q}) = \frac{\boldsymbol{q}}{\Delta t} - \boldsymbol{\mathcal{O}}$
      Construct Jacobian: $\boldsymbol{J} = \frac{1}{\Delta t}\boldsymbol{I} - \begin{bmatrix} -3G\frac{\partial\dot{\varepsilon}_{vp}}{\partial\sigma_e} & \frac{\partial\dot{\varepsilon}_{vp}}{\partial\rho_c} & \frac{\partial\dot{\varepsilon}_{vp}}{\partial\rho_w} \\ -3G\frac{\partial\dot{\rho}_c}{\partial\sigma_e} & \frac{\partial\dot{\rho}_c}{\partial\rho_c} & \frac{\partial\dot{\rho}_c}{\partial\rho_w} \\ -3G\frac{\partial\dot{\rho}_w}{\partial\sigma_e} & \frac{\partial\dot{\rho}_w}{\partial\rho_c} & \frac{\partial\dot{\rho}_w}{\partial\rho_w} \end{bmatrix}$
      Update: $\boldsymbol{q} \leftarrow \boldsymbol{q} - \frac{\boldsymbol{r}(\boldsymbol{q})}{\boldsymbol{J}}$
      Recompute Outputs/Rate vector with updated input, $\boldsymbol{\mathcal{I}}^{n+1}$
    end while
    Update State Variables:

$$\varepsilon_{vm} = \varepsilon^{old}_{vm} + \Delta p$$
$$\rho_c = \rho^{old}_c + \Delta\rho_c$$
$$\rho_w = \rho^{old}_w + \Delta\rho_w$$

  end if
  Compute Incremental Viscoplastic Strain Tensor: $\Delta\boldsymbol{\varepsilon}^{vp} = \frac{3}{2}\Delta p\,\frac{\boldsymbol{S}^{tr}}{\sigma^{tr}_e}$
  Compute Final Stress: $\boldsymbol{\sigma} = \boldsymbol{\sigma}_{old} + \boldsymbol{C} : (\Delta\boldsymbol{\varepsilon}_{total} - \Delta\boldsymbol{\varepsilon}^{vp})$
end for

2.2. Verification of Implementations

The proposed algorithms, which incorporate the surrogate elasto-viscoplastic constitutive model, have been thoroughly verified, and the corresponding results are outlined in this section. A total of nine verification cases were considered, each defined by distinct initial and boundary conditions, spanning the full stress–temperature input domain of the surrogate. These conditions were carefully chosen from the VPSC simulations originally employed to generate the surrogate's training dataset and are summarized in Table 1. In every case, a neutron flux, Φ, of 1×10^{-9} dpa/sec was prescribed. The MOOSE simulation data were obtained by modeling a single eight-noded hexahedral (HEX8) element subjected to uniaxial constant stress at a specified temperature. These simulation outcomes were directly compared against both the surrogate model predictions and the reference VPSC simulation results. The MoE surrogate responses were determined through direct time-domain integration of the model using the `scipy.integrate.solve_ivp` function with the `LSODA` solver option. `LSODA` serves as a Python interface to the Fortran ODEPACK package [21], which adaptively alternates between the non-stiff Adams method and the stiff backward differentiation formula, as originally introduced in [22]. Furthermore, the time evolutions of dislocation densities derived from the MOOSE simulations were systematically benchmarked against those predicted by the MoE surrogate model.

Table 1. Initial and boundary conditions used for the verification cases.

Case	**Effective Stress**	**Temperature**	**Effective Strain**	**Dislocation Densities**	
				at Cell	**at Cell Wall**
#	$\sigma_e\ (MPa)$	$T\ (K)$	ε_{vm}	$\rho_c\ (m^{-2})$	$\rho_w\ (m^{-2})$
1	50	683	2.1590×10^{-14}	5999999999923.7	9999999999985.1
2	50	850	1.1001×10^{-11}	5999999980883.5	9999999964211.5
3	50	1016	3.7299×10^{-9}	5999998717975.1	9999981465412.1
4	150	683	1.2609×10^{-11}	5999999994152.9	9999999992757.3
5	150	850	1.2609×10^{-11}	5999999994152.9	9999999992757.3
6	150	1016	3.7874×10^{-6}	6000005571349.8	9997755861825.3
7	250	683	5.6498×10^{-9}	5999999622770.7	9999996865525.4
8	250	850	2.7214×10^{-8}	5999998968779.3	9999984768044.6
9	250	1016	1.2765×10^{-7}	5999997505542.9	9999927814437.0

The verification results for the nine cases listed in Table 1 are shown in Figures 1, 2, and 3, in the same order in which they are listed in Table 1. It must be noted that presently the single measure of success for our implementations is how closely the MOOSE responses match the MoE surrogate responses. The observed deviations from the reference VPSC results are entirely attributed to the ML model, not to the algorithms proposed in this work. As mentioned earlier, the development and verification of the developed ML surrogate are presented in a separate work [14]. The presented verification plots demonstrate that the MOOSE responses are commensurate with the surrogate responses for all cases except Case 9.

Case 9 represents a combination of the highest stress and temperature of all the scenarios considered, and it is important to note that the surrogate model integration response using `scipy.integrate.solve_ivp` fails for Case 9. Although the MOOSE response does not entirely fail, and it provides a response that is closer to the reference solution, its error is significant. We present this case to emphasize that there are bounds to the stress and temperature regime that is applicable to this model. Even though the MOOSE responses might seem reasonable, in regions of extremely high stress and temperature, the results cannot be trusted. Additional data under those conditions will be needed to retrain the ML model to expand its region of applicability.

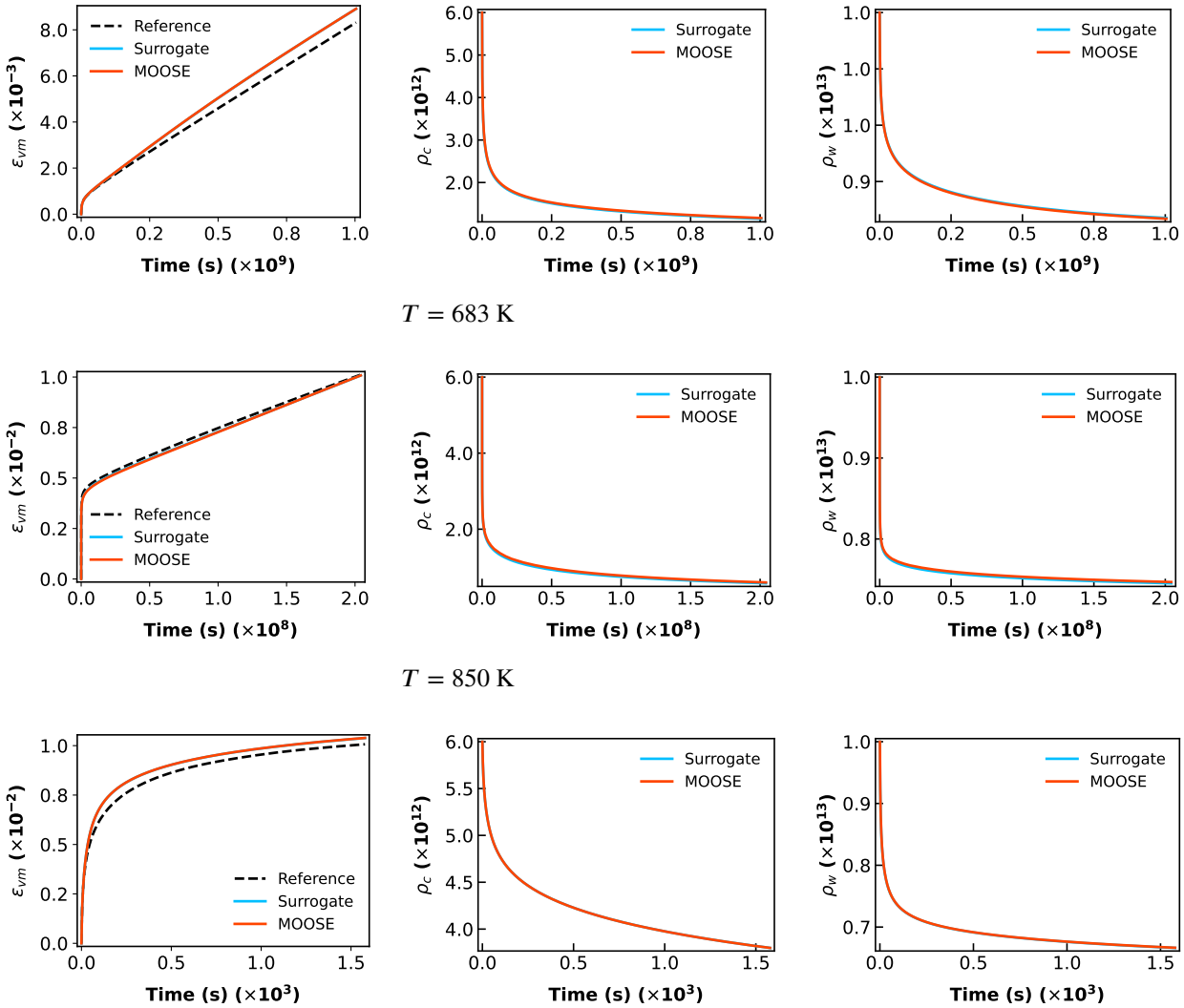


Figure 1. Comparison between the SciPy time integration of the surrogate (—) and the one-element (HEX8) response of the MOOSE (—) implementation for $\sigma_{vm} = 50$ MPa at different temperatures. The reference (- -) represents the VPSC data used for the MoE surrogate training.

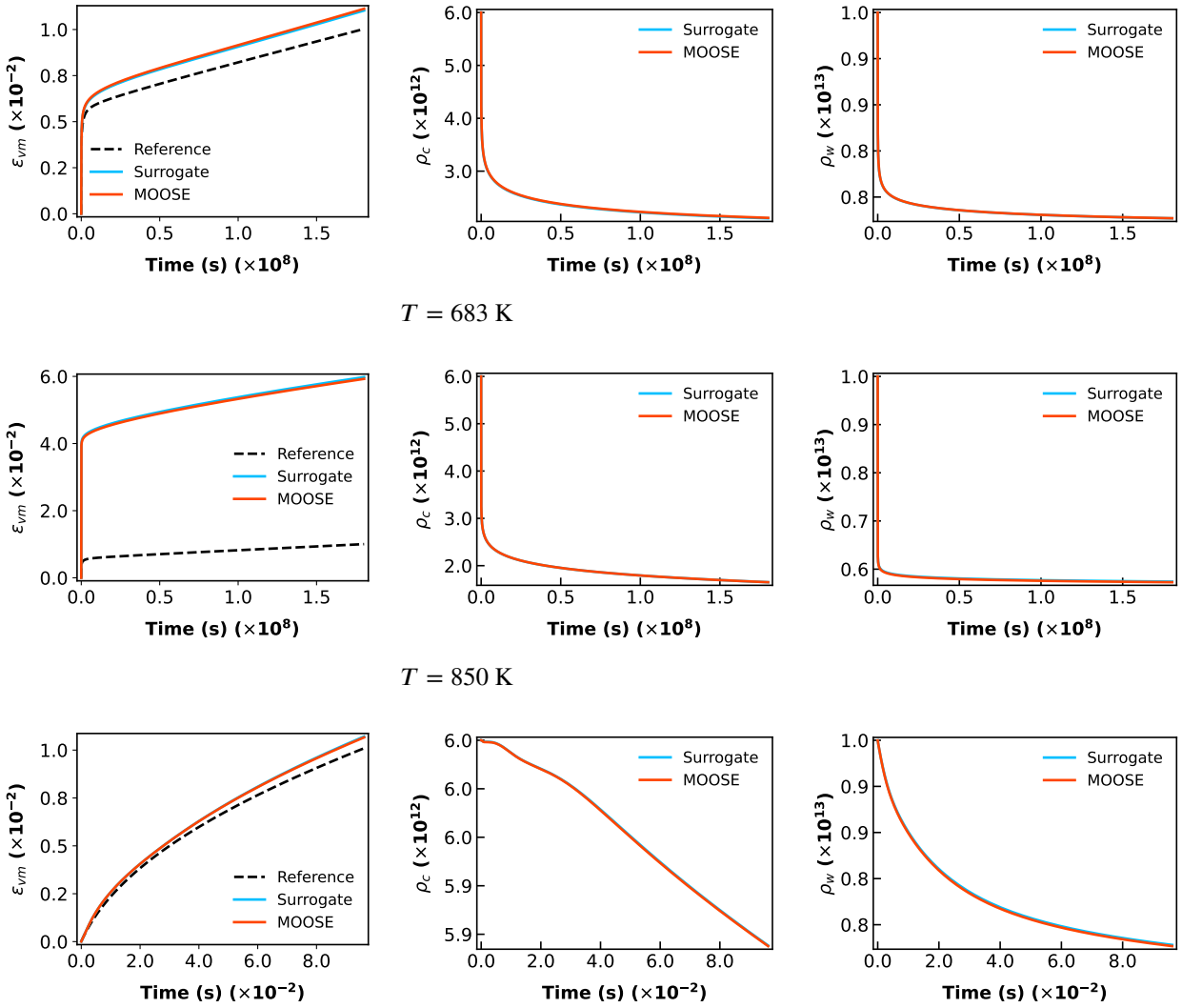


Figure 2. Comparison between the SciPy time integration of the surrogate (—) and the one-element (HEX8) response of the MOOSE (—) implementation for $\sigma_{vm} = 150$ MPa at different temperatures. The reference (- -) represents VPSC data used for the MoE surrogate training.

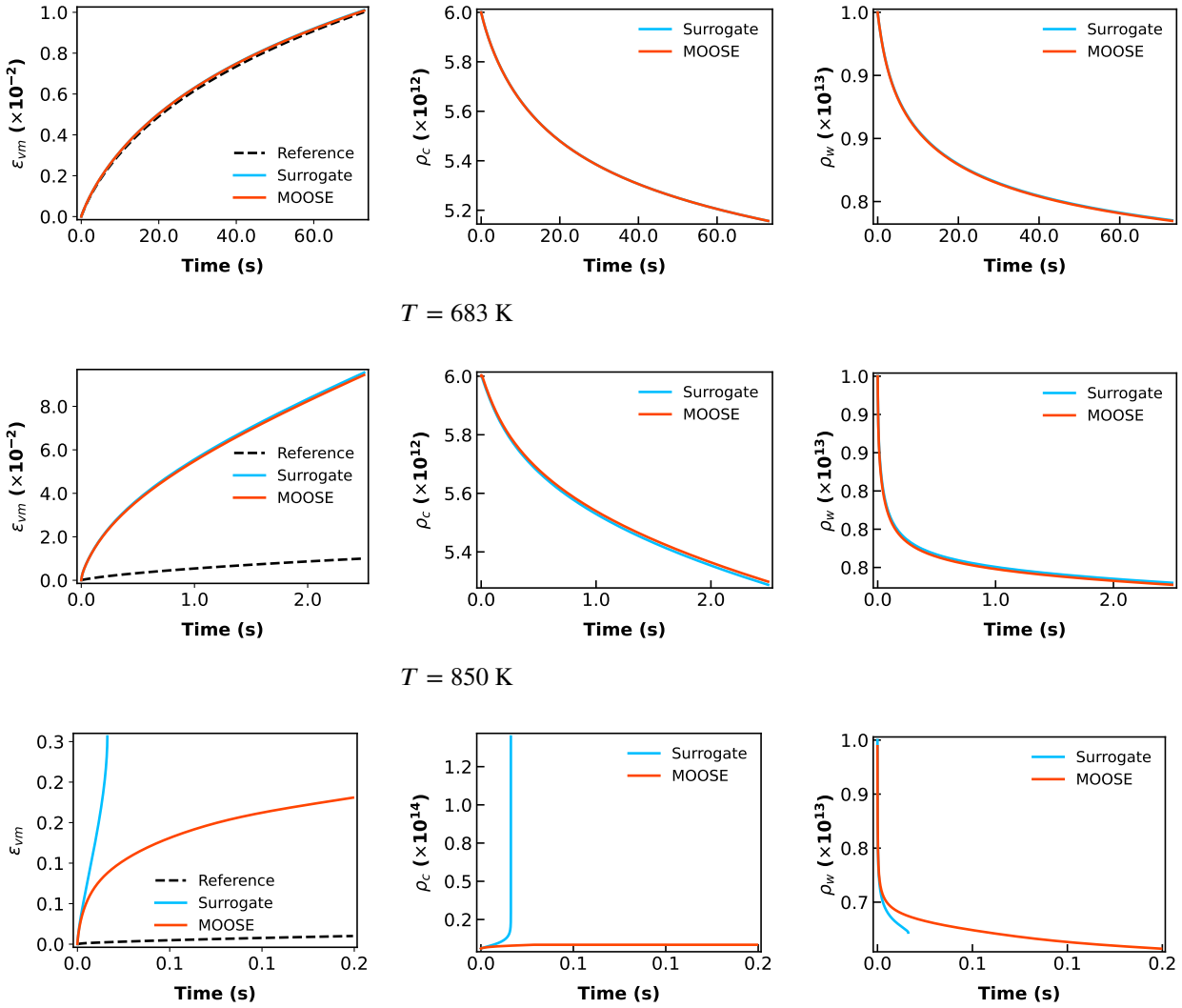


Figure 3. Comparison between the SciPy time integration of the surrogate (—) and the one-element (HEX8) response of the MOOSE (—) implementation for $\sigma_{vm} = 250$ MPa at different temperatures. The reference (- -) represents the VPSC data used for the MoE surrogate training.

2.3. Comparison Between the Univariate and Multivariate Schemes

This section summarizes an analysis of the accuracy of MNR-RR for the implicit integration proposed in this work, as compared to UNR-RR, which was used by the MoE surrogate model's predecessors [19, 20] and previous investigations of the current MoE model [12]. By design, MNR-RR is, mathematically speaking, expected to be more accurate than UNR-RR. On the other hand, solving for three quantities of interest (QoIs) is computationally more expensive than solving for only one. However, as will be made evident by the subsequent results and discussion, in most cases the difference in accuracy is too significant to ignore. Moreover, the computational cost associated with MNR-RR can be partially—if not completely—offset by the use of the larger time-step size enabled by MNR-RR during the initial load ramp.

In Figure 4, the left column presents the evolution of the elasto-viscoplastic strain and dislocation densities, as compared to the true evolution calculated by directly integrating the surrogate using the `scipy.integrate.solve_ivp` function at 160 MPa and 750 K with constant time-step sizes. In these investigations, the load was applied at the first time step and kept constant throughout, without any ramping up. Given that a converged solution is obtainable at a particular time-step size, the time step was also kept constant throughout. Figures 4(a), (c), and (e) clearly show that a meaningful converged solution can be obtained at a time-step size of up to 14 seconds when using the MNR-RR scheme. Although a negligible time discretization error could be observed (as evidenced by the difference between solutions using 5- and 14-second time steps), the trend is always maintained. However, the first converged solution obtained at a time-step size of 5 seconds using UNR-RR is grossly wrong. While a better solution could be obtained by reducing the time-step size to 1 second, that solution is unrepresentative of the actual one. Using the UNR-RR scheme, a representative and physically meaningful result is not obtained until decreasing the time-step size to 0.01 second. Even then, a significant discrepancy is observed in Figure 4(c) in the evolution of mobile dislocations in comparison to the actual surrogate response indicated by "Model." This is due to the fact that we are not actually solving for dislocation densities in the first place with UNR-RR, but rather updating them explicitly. In essence, the left column in Figure 4 shows that, compared to UNR-RR, MNR-RR can provide an accurate solution with a $14/0.01 = 1,400$ times larger time-step size thanks to its intrinsically higher level of accuracy.

Next, we determined the difference in overall accuracy under an adaptive time-stepping scheme in which the time step was adjusted based on nonlinear iteration counts in the nonlinear FE problem for a stress of 200 MPa at 850 K, shown in the right-hand column of Figure 4. Here, both the schemes were started from an initial small time step of 0.005 second and were allowed to grow until reaching the maximum time step size of 100 seconds. While the MNR-RR results exactly match the surrogate response, a stark difference is observed for UNR-RR. Although the elasto-viscoplastic strain evolution in Figure 4(b) approaches the true solution over time, it is easy to imagine that it could again diverge from the true solution over a longer simulation time. Much larger discrepancies are observed for the dislocation densities shown in Figure 4(d) and (f). These discrepancies arise due to the explicit updates contributing to the deviation of the elasto-viscoplastic strain from the true solution path, despite the fact that we are solving for it implicitly with the UNR-RR approach.

However, the more traditionally used scalar-valued UNR-RR is not without merit. At this point of our discussion, it is very apparent that the accuracy of the UNR-RR scheme is highly dependent on its time-step size. The smaller the time-step size, the lower the error introduced to the dislocation updates. In turn, this contributes to less deviation from the true solution. Therefore, at higher stress and temperature regimes, where the nonlinearity of the model outputs is extremely high, both schemes require an exceedingly small time-step size to converge within the primary creep regimes. In such a situation, the differences between the UNR-RR and MNR-RR responses vanish, and the UNR-RR approach proves itself beneficial thanks to its low computational overhead. In one such case, previously shown in Figure 3, the cases for all temperatures were

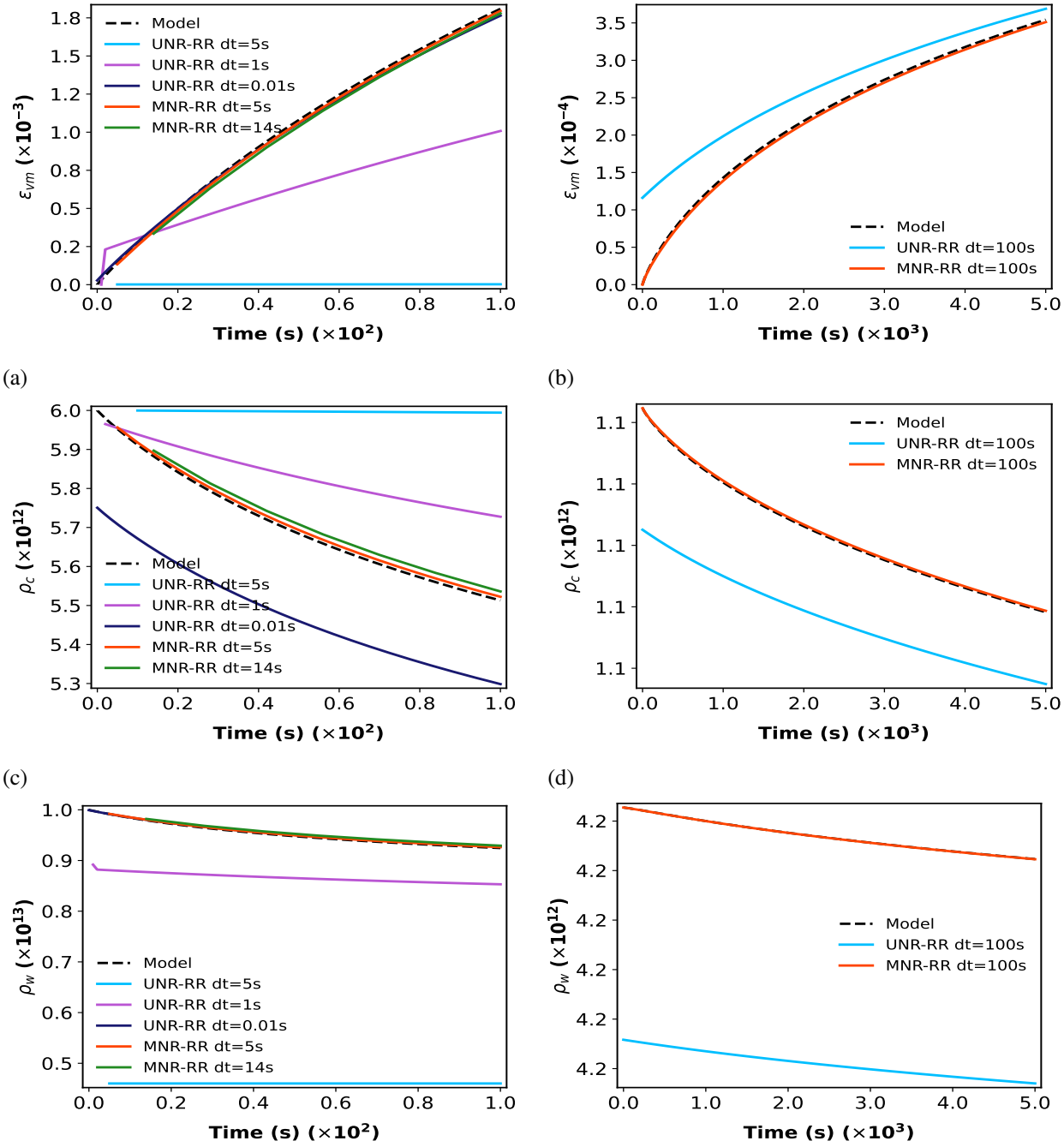


Figure 4. Comparison of the MNR-RR, UNR-RR, and baseline surrogate model responses (integrated with `scipy.integrate.solve_ivp`), with (left column) constant time steps and (right column) adaptive time steps restricted to 100 seconds.

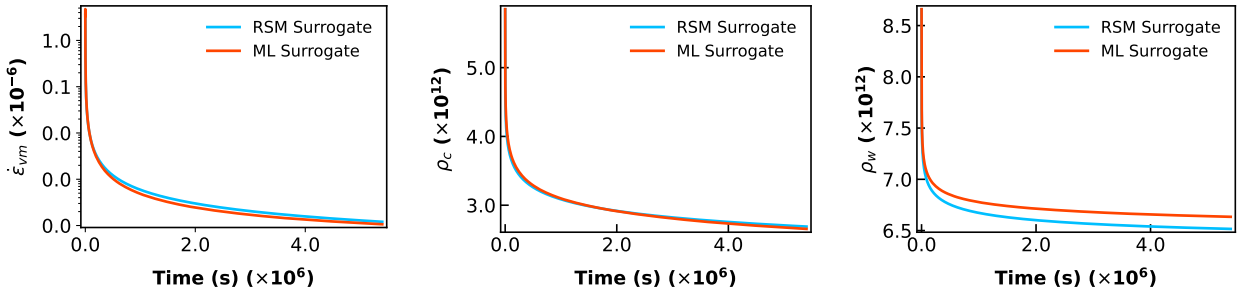
simulated using UNR-RR, and no appreciable difference from the true solutions was observed for either 683 or 850 K.

2.3.1. Performance Analyses

While the implemented ML model's input space covers a much larger range of stress and temperature than does its predecessor reduced order model (ROM) implementation `HT9LaRomance` (available in the MOOSE-based fuel performance code BISON and developed using RSM [20]), it also outperforms the previous ROM within their common input space in terms of prediction accuracy and reliability, and its predictions are physically reasonable based on the mechanisms governing creep deformations at high temperature and/or stress. Comparisons will be made here both in the primary and secondary creep regions. However, before discussing the scenarios in which the MoE surrogate surpasses the RSM surrogate, we provide a base case to establish that both models provide more or less similar responses for the initial conditions at which they are known to perform well. To this end, a common set of initial and boundary conditions, shown in Table 2, was used to compare both models. The corresponding responses are shown in Figure 5. Both models provide similar responses. The negligible discrepancies are due to slight differences between the VPSC datasets on which the models were trained. Note that the y-axis of the strain rate presented in Figure 5(a) is plotted in logarithmic scale to better amplify the negligible response differences.

Table 2. Initial and boundary conditions for the base case.

Initial Conditions	Symbols	Values
Effective stress	σ_e	128.7 MPa
Temperature	T	789 K
Effective strain	ε_{vm}	0.0
Dislocation densities at cell	ρ_c	$5.85 \times 10^{12}\ m^{-2}$
Dislocation densities at cell wall	ρ_w	$8.66 \times 10^{12}\ m^{-2}$
Neutron flux	Φ	3.82×10^{-8} dpa/s



(a) (b) (c)

Figure 5. Comparison of RSM and ML surrogate model predictions of (a) effective strain rate and dislocation densities at the (b) cell and (c) cell wall under a stress of 127.8 MPa and at 789 K, simulated for 1,500 hours.

2.3.1.1. *Primary Creep Rates*

One limitation of the previous RSM-based surrogate is that it performs poorly in the transition regions where the dominant deformation mechanism changes from one creep mechanism to another. This is evident from the primary creep rate presented in Sweet and Novascone [19]. The irregularity in the primary creep strain rates is due to the presence of regions in which the creep rates do not increase with increasing stress or temperature—as they generally should do as long as the mechanisms and physics behind the deformations are in effect—and vice versa. To demonstrate this, we chose a temperature of 810 K, which is well within the transition region where the activation energy of the creep mechanism transitions from the activation energy for the grain boundary diffusion, Q_{GB}, to the activation energy for the lattice diffusion, Q_L. Generally, our choices of applied stress were not high enough to free up the dislocations altogether to fully change the creep mechanism. However, for increasing stress at this constant temperature, the dominant creep mechanism shifts from the point defect diffusion-based viscous creep toward the dislocation-based power-law creep. Figure 6 shows the results of the comparison between the RSM and implemented ML model for constant uniaxially applied stresses ranging from 40–70 MPa. All cases were simulated for 20 days to capture the trends in primary creep rates only. For the RSM case in Figure 6(a), the aforementioned anomaly is easily observed. While the primary creep rate does increase for 50 MPa compared to 40 MPa, it decreases unreasonably—and perhaps nonphysically—for 60 and 70 MPa. On the other hand, in Figure 6(b), the primary creep rates increase with increasing stress for the MoE model, clearly demonstrating the improved behavior of the ML surrogate compared to the previous ROM in handling transitions.

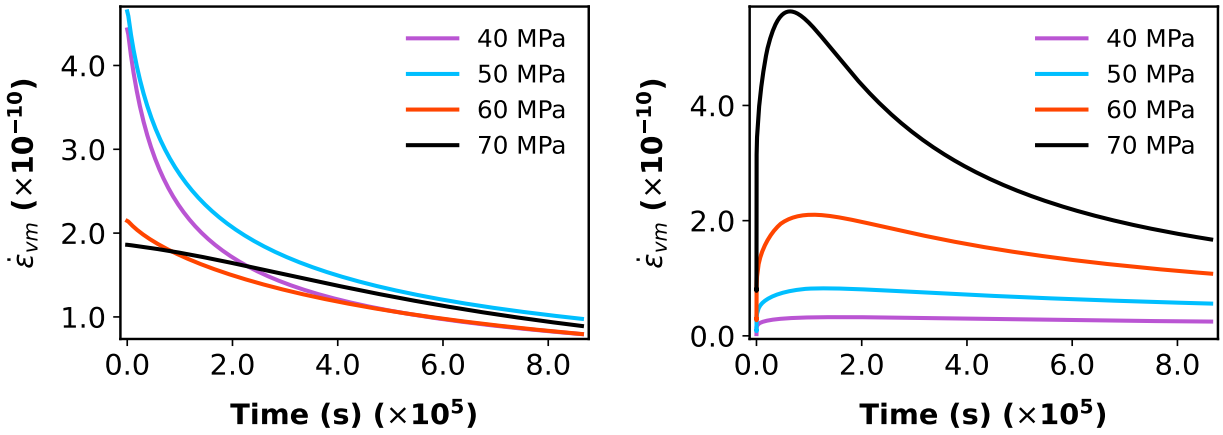


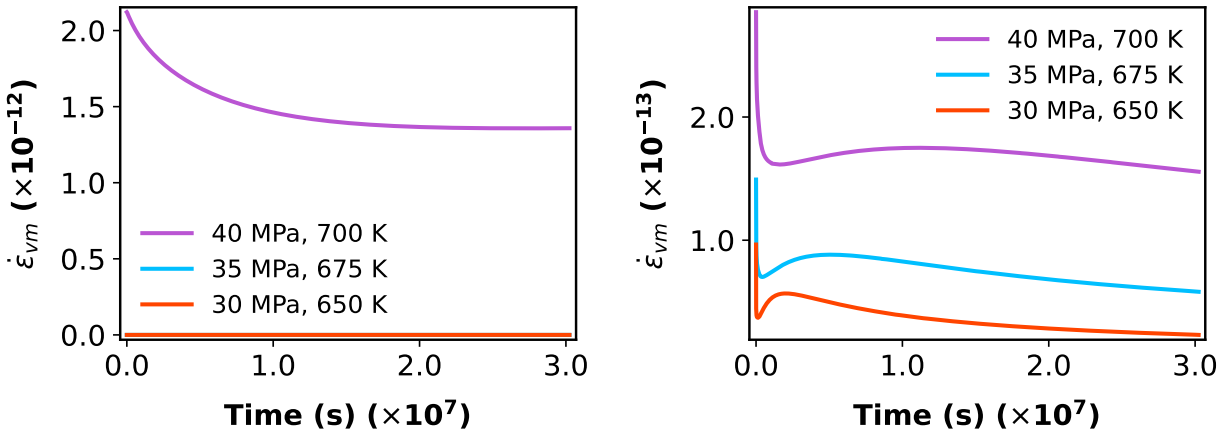
Figure 6. (a) RSM and (b) ML surrogate model primary creep rates for a range of applied stress at 810 K, simulated for 20 days.

2.3.1.2. *Secondary Creep Rates*

Similarly, several irregularities were previously observed for secondary creep rates over the ROM feature space for the RSM surrogate. The most prominent was a sudden drop in creep rate to almost zero for uniaxial tensile cases below a stress of 40 MPa and a temperature of 700 K [19]. This phenomenon is corroborated in the RSM's secondary creep predictions, as shown in Figure 7(a), while the corresponding results from the ML surrogate model in Figure 7(b) show that the ML surrogate overcomes these issues. All three cases were

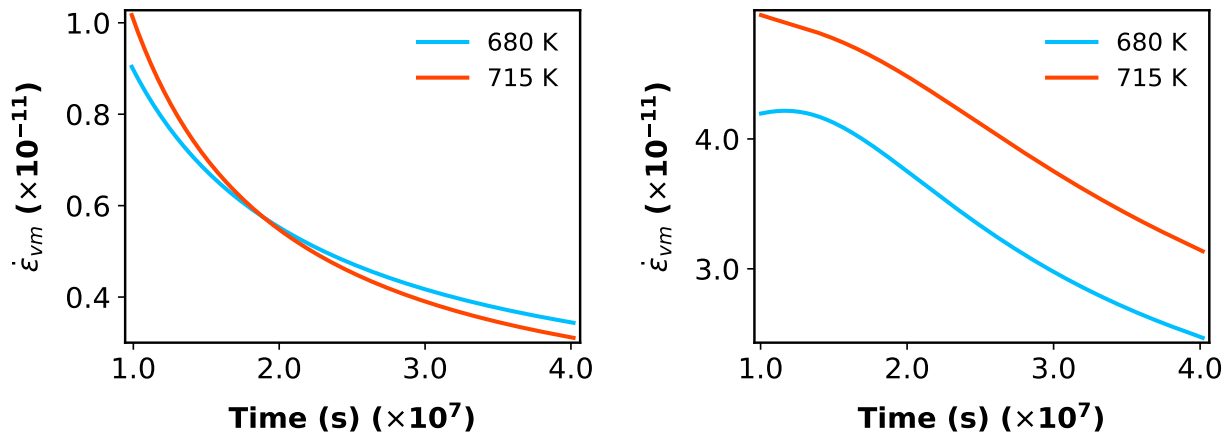
simulated for 350 days, which ensured that a steady-state creep regime was reached.

Well within the steady-state region, anomalies in handling transitions between creep mechanisms are also prevalent for the RSM. This is shown in Figure 8, which shows the results of a simulation in which a stress of 115 MPa was applied; this stress was selected to fall within the transition region of viscous creep to power-law creep. Under this constant stress, the temperature was raised from 680 to 715 K to change the activation energy governing the dominant creep mechanism. Within the secondary creep regime, the creep strain rate of the higher-temperature case was found to unrealistically drop below the lower-temperature case. This is evident in the crossing of the plots of the effective creep rates predicted at 680 and 715 K, shown in Figure 8(a), in which only the secondary creep regimes that were simulated for 450 days are shown to highlight this behavior. On the other hand, in Figure 8(b), the implemented MoE model's response shows reasonable and consistent predictions; it maintains more physically realistic trends in creep rates without the crossover behavior observed for the RSM.



(a) (b)

Figure 7. Comparison of secondary creep rates between the (a) RSM and (b) ML surrogate model simulated for 350 days for stresses and temperatures at and below 40 MPa and 700 K.



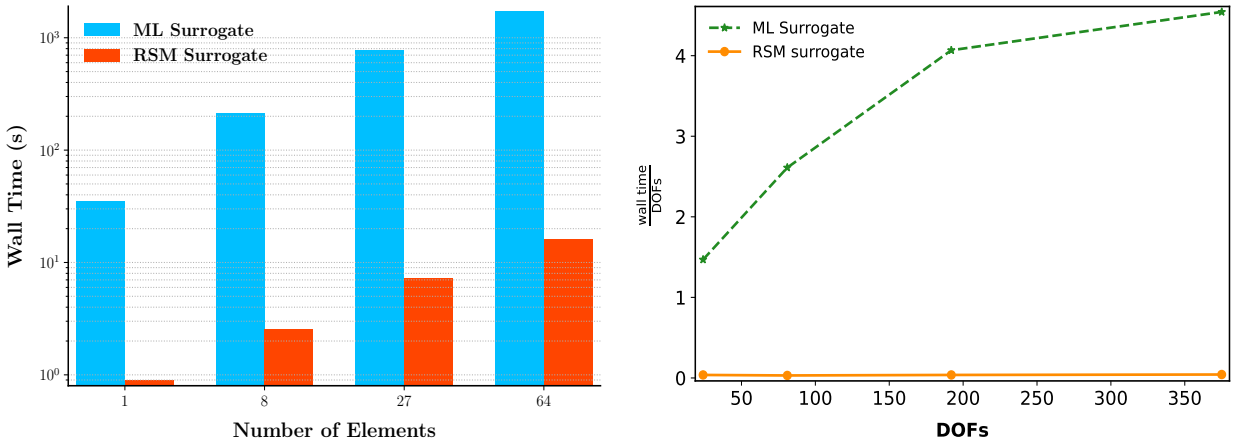
(a) (b)

Figure 8. Comparison of secondary creep rates between the (a) RSM and (b) ML surrogate model simulated for 450 days for a stress of 115 MPa and temperatures of 680 K and 715 K.

2.3.1.3. Computational Overhead

While the ML-based constitutive laws and flow rates have been shown to offer improved accuracy over a feature space encompassing multiple scales and mechanisms, their applications to realistic problem simulations are scarce due to the prohibitively large computational overhead associated with them. Naturally, closed-form mathematical expressions are many times faster than a trained neural network with numerous intrinsic computations. Depending on the context and operational requirements, in many applications the benefits of improved accuracy can easily offset the higher computational times required. Therefore, we compared the RSM surrogate against the MoE ML surrogate to gain insights into the related computational costs. The comparison was made for a stress of 150 MPa and a temperature of 683 K across an increasing number of elements under uniaxial tensile loading. The results are presented in Figure 9. All cases were simulated for 20 days.

Figure 9(a) shows the wall time (with the y-axis using a log scale) as a function of the number of elements, while Figure 9(b) shows how each model fares when the model size is increased (and the number of degrees of freedom (DOFs) is scaled up). As expected, the ML model requires considerably more computational time than the RSM. Notably, even the most optimal variant of the ML model, which utilized the "trace" method in the libTorch package it was developed in, falls behind the ROM in terms of evaluation speed. Moreover, more overhead is added as the number of elements increases, although Figure 9(b) clearly shows that the added per-DOF overhead saturates and plateaus with increasing DOFs for the ML surrogate. Performance of an ML surrogate integrated into an FE framework is expected to be improved through vectorization of all the quadrature points and simultaneous computation through a graphics processing unit. Moreover, further pruning the MoE surrogate by removing unnecessary weights and neurons for improved optimization is expected to lead to faster inference with reduced overhead.



(a) (b)

Figure 9. Comparison of the RSM and ML surrogate showing (a) wall time and (b) wall time per DOF for a stress of 150 MPa and a temperature of 683 K simulated for 20 days.

2.4. Summary and Future Work

In this report, we introduced two implicit integration schemes for incorporating the ML constitutive relation, designed to predict the elasto-viscoplastic response of HT9 steel within the MOOSE framework,

by adapting the RR method for iterative stress correction. In both formulations, the viscoplastic strains were updated implicitly, but the schemes are distinguished as univariate and multivariate depending on the procedure used for updating dislocation densities. The newly proposed multivariate formulation exhibited superior accuracy relative to the univariate scheme, which has conventionally been employed to integrate its RSM-based predecessor. A comprehensive set of cases was examined, covering the entire input domain of the MoE surrogate, to rigorously verify the performance of the implementations. The FE responses of a single HEX8 element from MOOSE under uniaxial loading were benchmarked against both VPSC reference simulations and surrogate model predictions. The surrogate model responses were obtained by time integration using the well-established SciPy package [23]. The verification outcomes demonstrated excellent agreement, remaining within the surrogate's trusted domain. Moreover, we showed that the developed MoE ML surrogate and its implementations effectively address the known shortcomings of the earlier RSM-based surrogate and the FE implementation, `HT9LaRomance`.

Nevertheless, the verification study also revealed the breakdown of both the surrogate and the FE predictions at high-stress and high-temperature regimes, such as in Case 9. In this range, the strong nonlinearities induced by stress and temperature variations are considerably more difficult for the surrogate model to learn and accurately reproduce. Expanding the number of trainable parameters in the ML model may help mitigate this limitation. In addition, the relatively higher computational cost compared to its predecessor introduces a trade-off between predictive accuracy and computational efficiency in engineering-scale simulations. This issue, however, can be effectively resolved by vectorizing all quadrature points to leverage graphics processing unit acceleration and/or by optimizing the surrogate through model pruning. While the first challenge must be addressed during surrogate development, the second pertains to implementation. Both are important directions for future work.

3. XFEM DEVELOPMENT AND INTEGRATION WITH CREEP MODELS

In addition to being able to predict creep deformation in high-temperature alloys in nuclear applications, it is important to be able to predict the growth of cracks in those alloys. Cracks can nucleate in regions of local imperfections, potentially aided by corrosion processes, and then propagate in a process known as creep crack growth (CCG). It is important to ensure that postulated cracks or cracks detected during in-service inspections would propagate slowly enough that component integrity is not compromised. Creep deformation has a significant influence on CCG, and this project aims to utilize the surrogate creep model presented in the previous section to model creep deformation within CCG simulations.

The XFEM [24] is an attractive approach for representing fracture in engineering-scale fracture simulations within the FE method because it permits fractures to propagate in a manner that is independent of the underlying FE mesh and allows the asymptotic stress concentrations that occur near the crack tip to be modeled, and it incurs relatively little computational overhead in doing so. Because of these advantages of XFEM, this project is pursuing an approach where the surrogate creep model is used in conjunction with XFEM to model CCG.

In this section, we present our recent progress in CCG modeling using the aforementioned approach. The summary of this effort provided here is organized into five key components:

- Numerical integration of strain energy release density (SERD): We developed new implementations for computing SERD, which serves as the driving force for CCG. For simple creep laws, such as the power-law model, closed-form solutions for SERD integration are readily available. However, for more complex creep laws—particularly those represented by surrogate models lacking explicit formulations—analytical integration becomes infeasible, necessitating numerical approaches.
- Enhancement of XFEM in the MOOSE framework: We updated the current XFEM implementation in MOOSE to allow crack tips to be positioned arbitrarily within FEs using near-tip enrichment functions for inelastic models. The prior implementation restricted crack advancement to element edges, which is unsuitable for CCG simulations in which crack extension depends on the continuous evaluation of contour integrals derived from SERD.
- Automatic differentiation for robustness: We introduced a new implementation of near-tip enrichment using automatic differentiation, which significantly improves the robustness and accuracy of the XFEM code when using near-tip enrichment.
- Verification and validation: We verified the newly implemented features against analytical solutions and benchmark results, and validated them against experimental data to ensure accuracy and reliability.
- Integration of surrogate models: We demonstrated the capability to integrate surrogate models into the XFEM framework, enabling the simulation of CCG driven by complex, data-informed creep behavior.

3.1. XFEM Formulations

3.1.1. Enriched Solutions

In XFEM, at a location $\boldsymbol{x} \in R^d$, with d being the problem dimension, the approximation for the vector-valued displacement field $\boldsymbol{u}(\boldsymbol{x})$ is expressed as [25]

$$u_i^h = \sum_{I \in S_s} N_I(\boldsymbol{x}) u_{Ii} + \sum_{J \in S_{en}} N_J(\boldsymbol{x}) \sum_{K=1}^{4} M_K(\boldsymbol{x}) u_{KJi}, \tag{11}$$

where S_s is the set of standard FE nodes, S_{en} is the set of enriched nodes surrounding a crack tip, and i is the spatial index ranging from 1 to d. $N(\boldsymbol{x})$ denotes the standard FE shape functions, while the enriched shape functions $\boldsymbol{M}(\boldsymbol{x})$ are defined as

$$\boldsymbol{M}(\boldsymbol{x}) = \{r^{\frac{1}{n}} \cos\frac{\theta}{2}, r^{\frac{1}{n}} \sin\frac{\theta}{2}, r^{\frac{1}{n}} \sin\frac{\theta}{2}\sin\theta, r^{\frac{1}{n}} \cos\frac{\theta}{2}\sin\theta\}, \tag{12}$$

where r is the radial distance from the crack tip, θ is the angle relative to the local crack direction, and n is an exponent that depends on the material behavior. For linear elasticity, $n = 2$ is typically used and is consistent with the degree of singularity observed in that case.

3.1.2. Governing Equations

Consider a solid body $B \in R^d$. The deformation of a material point at $\boldsymbol{x} \in B$ is described by the displacement field $\boldsymbol{u}(\boldsymbol{x}) : R^d \rightarrow R^d$. The governing equations for material deformation are given by

$$\nabla \cdot \boldsymbol{\sigma} = \boldsymbol{0} \qquad \text{in } B, \tag{13a}$$

$$\boldsymbol{u} = \bar{\boldsymbol{u}} \qquad \text{on } \partial B_u, \tag{13b}$$

$$\boldsymbol{\sigma} \cdot \boldsymbol{n} = \bar{\boldsymbol{t}} \qquad \text{on } \partial B_t, \tag{13c}$$

where $\partial B_u \cap \partial B_t = \emptyset$ and $\partial B_u \cup \partial B_t \cup \partial B_c = \partial B$. The weak form of Equation (13) is obtained by multiplying with a test function $\dot{\boldsymbol{u}}$, yielding

$$(\dot{\boldsymbol{\epsilon}}, \boldsymbol{\sigma})_B - \left(\dot{\boldsymbol{u}}, \bar{\boldsymbol{t}}\right)_{\partial B_t} = 0, \tag{14}$$

where $(\cdot,\cdot)_B$ denotes the integral over the domain B and the strain tensor $\boldsymbol{\epsilon}$ in the small strain setting is given by $\epsilon_{ij} = 0.5(u_{i,j} + u_{j,i})$. Discretizing Equation (14) using the approximation from Equation (11) for both $\boldsymbol{u}$ and $\dot{\boldsymbol{u}}$, the residual vector is given by

$$R_a = \left(B_{I,ijk}, \sigma_{jk}\right)_B - \left(N_I, t_i\right)_{\partial B_t}, \tag{15}$$

where $B_{I,ijk}$ represents the strain-displacement matrix components at node I. The nodal solution u_b is obtained through an iterative update,

$$K_{ab} = \frac{\partial R_a}{\partial u_b}, \tag{16a}$$

$$\boldsymbol{u}^{n+1} = -\boldsymbol{R}\boldsymbol{K}^{-1} + \boldsymbol{u}^n, \tag{16b}$$

where $a = Ii + 4Ji$ is the degree-of-freedom index. For automatic differentiation (AD) implementations, the explicit form of the stiffness matrix $\boldsymbol{K}$ in Equation (16a) is not required to be implemented by the developer, as it is computed through AD operations on the residual R_a with respect to the nodal values u_b [26]. Details of the non-AD implementation of $\boldsymbol{K}$ can be found in Sukumar et al. [24].

3.1.3. J-Integral

The J-integral is commonly used in fracture mechanics to quantify the energy release rate associated with crack growth. It provides a path-independent measure of the intensity of the stress and strain field near the crack tip, making it a valuable tool for evaluating crack driving forces in both linear and nonlinear materials

[27]. Given the normal vector $\boldsymbol{n}$ to the crack surface Γ, oriented along the $\hat{x} - axis$ of a local Cartesian coordinate system, the J-integral is defined as

$$J = \lim_{\Gamma \to 0} \int_{\Gamma} T_{\hat{1}\hat{a}} n_{\hat{a}} d\Gamma, \tag{17}$$

where the Eshelby tensor $T_{\hat{1}j}$ is given by

$$T_{\hat{1}j} = w^e \delta_{\hat{1}j} - \sigma_{ij} u_{i,\hat{1}}, \tag{18}$$

with w^e denoting the strain energy density, $\delta_{\hat{1}j}$ the Kronecker delta, σ_{ij} the Cauchy stress tensor, and $u_{i,\hat{1}}$ the derivative of the displacement field in the local x-direction.

3.1.4. C-Integral

The $C(t)$-integral is introduced as a time-dependent counterpart to the classical J-integral to account for material behavior under creep conditions. While the J-integral is path-independent and widely used for characterizing crack tip fields in elastic and elastic-plastic materials under steady-state loading, it does not capture the time-dependent nature of deformation in viscoelastic or creep-dominated materials.

In contrast, the $C(t)$-integral incorporates the rate of energy dissipation due to creep through the SERD $\dot{w}^e$, making it suitable for problems where time-dependent inelastic deformation mechanisms are significant. The theoretical foundation for the $C(t)$-integral was extensively developed in the works of Riedel and Nikbin [28, 29]. The $C(t)$ integral is defined as

$$C(t) = \lim_{\Gamma \to 0} \int_{\Gamma} \dot{T}_{\hat{1}\hat{a}} n_{\hat{a}} d\Gamma, \tag{19}$$

where

$$\dot{T}_{\hat{1}j} = \dot{w}^e \delta_{\hat{1}j} - \sigma_{ij} \dot{u}_{i,\hat{1}}. \tag{20}$$

The strain rate energy density $\dot{w}^e$ in Equation (19) takes the form of

$$\dot{w}^e = \int_0^{\dot{\boldsymbol{\epsilon}}} \boldsymbol{\sigma} \, : \, \mathrm{d}\dot{\boldsymbol{\epsilon}}. \tag{21}$$

In the case of uniaxial monotonic loading, Equation (21) simplifies to

$$\dot{w}^e = \int_0^{\dot{\bar{\epsilon}}^{eq}} \bar{\sigma}^{eq} \, \mathrm{d}\dot{\bar{\epsilon}}^{eq}, \tag{22}$$

where $\bar{\sigma}^{eq} = \sqrt{\frac{3}{2} \boldsymbol{s} \, : \, \boldsymbol{s}}$ is the effective stress with $\boldsymbol{s} = \boldsymbol{\sigma} - \frac{1}{3}(\boldsymbol{\sigma} \, : \, \boldsymbol{I})\boldsymbol{I}$ being the deviatoric stress tensor, and $\dot{\bar{\epsilon}}^{eq} = \sqrt{\frac{2}{3} \dot{\boldsymbol{\epsilon}} \, : \, \dot{\boldsymbol{\epsilon}}}$ is the effective strain rate.

3.1.4.1. Incremental Evaluation

The integral in Equation (21) can be numerically calculated using an incremental form,

$$w^e_{n+1} = w^e_n + \Delta w^e, \tag{23}$$

where the energy density increment is defined as

$$\Delta w^e = \frac{1}{2}(\boldsymbol{\sigma}_{n+1} + \boldsymbol{\sigma}_n) : \frac{(\boldsymbol{\epsilon}_{n+1} - \boldsymbol{\epsilon}_n)}{\Delta t}. \tag{24}$$

This incremental formulation is general and applicable to any constitutive relation between stress and strain. It is particularly advantageous for integrating surrogate models into FE codes, where closed-form expressions for energy density may not be available. In such cases, the incremental approach becomes the only feasible option for implementation, as will be demonstrated in the next section.

3.1.4.2. Legendre-Based Evaluation

The incremental evaluation approach presented in Section 3.1.4.1 can introduce errors if time steps are large. Numerically integrating the integral in Equation (22) can minimize those errors. The documentation for the Abaqus code [30] indicates that a five-point Gauss rule is used in that code to numerically integrate that equation, but doing so can be challenging for standard creep models that express the creep rate as a function of the stress because that form requires stress as a function of strain.

We address this challenge by evaluating the integral in Equation (22) using a Legendre transformation. By applying integration by parts, Equation (22) becomes

$$\dot{w}^e = \bar{\sigma}^{eq}\dot{\bar{\epsilon}}^{eq} - \int_0^{\bar{\sigma}^{eq}} \dot{\bar{\epsilon}}^{eq}(\bar{\sigma}^{eq})\,\mathrm{d}\bar{\sigma}^{eq}, \tag{25}$$

where $\bar{\sigma}^{eq}$ and $\dot{\bar{\epsilon}}^{eq}$ are the equivalent stress and equivalent strain rate, respectively. The Legendre-based formulation in Equation (25) requires an explicit expression for $\dot{\bar{\epsilon}}^{eq}$ as a function of $\bar{\sigma}^{eq}$. Its accuracy depends solely on the number of integration points used, which makes it a preferred method in practice when such a relation is available.

We employ the following approach for numerically evaluating the integral using the numerical integration approach. The second term in Equation (25), denoted as $\dot{\tilde{w}}^e$, is computed numerically as

$$\dot{\tilde{w}}^e = \int_0^{\bar{\sigma}^{eq}} \dot{\bar{\epsilon}}^{eq}(\bar{\sigma}^{eq})\,\mathrm{d}\bar{\sigma}^{eq} \tag{26}$$

using Gaussian quadrature. The procedure is as follows:

$$\bar{\sigma}_i^{eq} = 0.5(1 + \xi_i)\bar{\sigma}^{eq}, \tag{27a}$$

$$\boldsymbol{J} = 0.5\bar{\sigma}^{eq}, \tag{27b}$$

$$\dot{\tilde{w}}^e = \boldsymbol{J}\sum_{i=1}^{nip} \dot{\bar{\epsilon}}^{eq}(\bar{\sigma}_i^{eq})w_i, \tag{27c}$$

where ξ_i and w_i are the quadrature points and weights, and nip is the number of integration points. Based on our numerical experience, using $nip = 5$ provides sufficient accuracy for most applications.

In our MOOSE-based code design, there are three options for computing the SERD $\dot{w}^e$ in Equation (22). The default option is to compute $\dot{w}^e$ directly if a closed-form expression is available—for example, in the case of a power-law creep model. Otherwise, numerical integration is performed using either the formulation in Section 3.1.4.2 or, by default, the one in Section 3.1.4.1.

3.1.5. Creep Growth Rate

Following Kumar and Singh [31] and Anderson [32], a Paris law-type relation is utilized to describe the crack growth rate as a function of the $C(t)$-integral,

$$\frac{\Delta a}{\Delta t} = B[C(t)]^m, \tag{28}$$

where Δa is the crack extension over a time increment Δt, B is the crack growth coefficient, and m is the material-dependent exponent that characterizes the nonlinearity of the crack growth behavior. Both m and B are considered material properties and must be calibrated against experimental data.

This CCG model was integrated into the MOOSE XFEM capability by using the mesh-based cutter, the development of which is described in Bajpai et al. [33]. For 2D problems, a set of topologically 1D cutting meshes are used to represent individual cracks, and the cutting mesh is incrementally extended as the cracks propagate. For this application, the crack propagation increment is computed based on the Paris law for CCG.

3.2. Numerical Results

In this section, a series of 2D numerical examples is presented that verify and demonstrate the capabilities of the newly developed code. First, we present the testing of the contour integral, for which a simple case of linear elasticity was used to evaluate the J-integral. Next, we cover the verification of the computation of the $C(t)$ integral using a power-law creep model. This is followed by a linear elasticity example of crack propagation, which illustrates the effect of crack-tip enrichment on the computed J-integral. A similar demonstration was extended to the $C(t)$ integral in a CCG simulation. Finally, the last example demonstrates the feasibility of integrating the MoE model from Section 2 into the XFEM framework.

3.2.1. Verification for a Stationary Crack

3.2.1.1. Linear Elastic Material

A double-notched plate was considered, with a uniform vertical pressure of $\bar{\sigma} = 100\,\text{MPa}$ applied to its top surface. Due to geometric symmetry, only half of the model was analyzed. The geometry and boundary conditions are illustrated in Figure 10. The material was assumed to be linear elastic, with a Young's modulus of $E = 207,000\,\text{MPa}$ and a Poisson's ratio of $\nu = 0.3$.

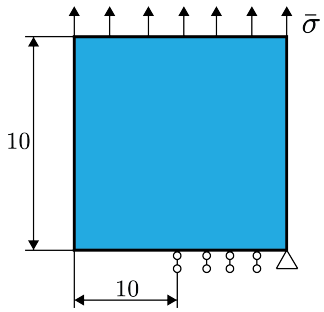


Figure 10. Geometry and boundary condition setup of the tensile test. All dimensions are in mm.

Figure 11 presents a convergence study of the J-integral as the mesh size decreases from $h = 1$ to

1/64 mm. The numerical solutions (in blue) converge toward the analytical solution (in red), with a reference value of 2.434 MPa · mm that was reported in Hinton and Ezatt [34].

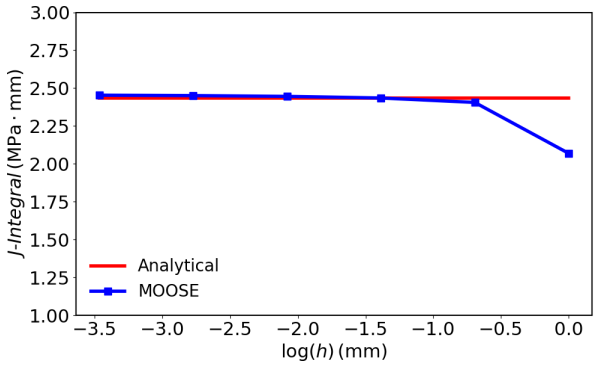
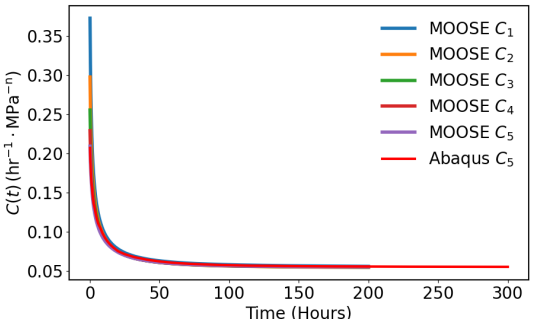


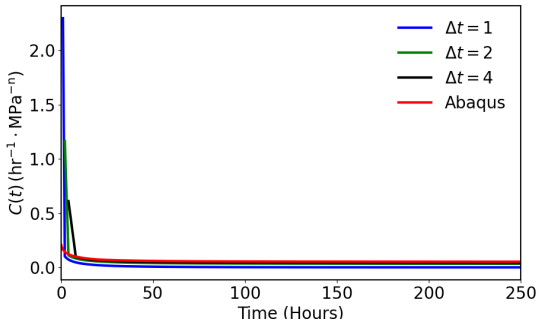
Figure 11. Mesh size convergence study for the J-integral computed from the tensile test.

3.2.1.2. Power-Law Creep Material

The stationary crack problem was repeated using the same geometry and boundary conditions described above; however, the material behavior was governed by a power-law creep model, defined as $\dot{\bar{\epsilon}}^{eq} = C(\bar{\sigma}^{eq})^n$, instead of linear elasticity. The computational domain was discretized using a mesh size of $h = 1/4$ mm. Using this setup with $C = 5 \cdot 10^{-12}\,\mathrm{hr}^{-1} \cdot \mathrm{MPa}^{-\mathrm{n}}$ and $n = 3$, the $C(t)$-integral was evaluated to assess time-dependent fracture behavior. We present results obtained using two numerical methods for evaluating the SERD, as described in Equations (25) and (24). These results are compared with those from Abaqus, which employs a five-point integration rule. Figure 12(a) compares the computed $C(t)$ values using different radii, based on the method described in Section 3.1.4.2, against the Abaqus reference. As shown, $C(t)$ values are sensitive to the chosen radii during the transitional phase from $t = 0$ to 50 hours. Eventually, the values converge to a steady state.



(a) Results from the five-point integration rule described in Section 3.1.4.2. Here, C_n denotes the contour area associated with the n-th layer surrounding the crack tip, where increasing n corresponds to regions progressively father from the tip.



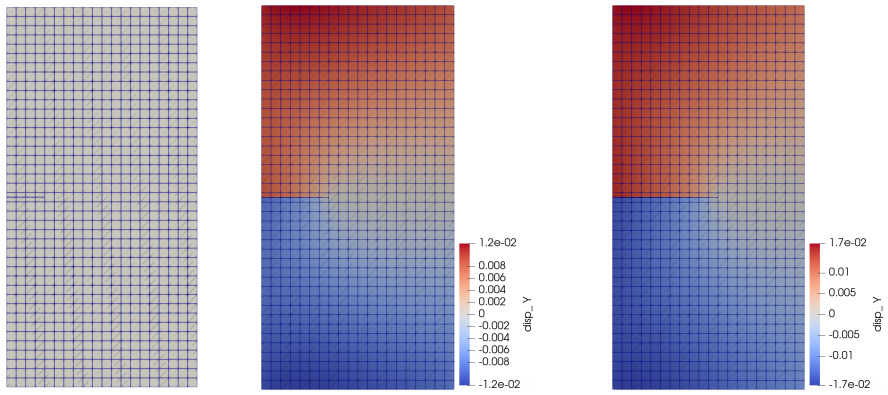
(b) Results from the incremental approach described in Section 3.1.4.1, utilizing the contour area C_5, which corresponds to the fifth layer away from the crack tip, with various time step sizes.

Figure 12. Numerical results from various integration schemes, contour sizes, and time steps for evaluation of the contour integral $C(t)$ for the tensile test with a creep model, compared with results from the Abaqus code.

Figure 12(b) compares the $C(t)$-integral as a function of the SERD $\dot{w}^e$, computed using the method in Section 3.1.4.1, with the Abaqus reference from Figure 12(a) and the same contour radii. The results indicate that $C(t)$ is slightly sensitive to the time step Δt. As the time step increases, the computed $C(t)$ time history approaches the Abaqus result in the secondary creep regime, while the primary creep region exhibits significant sensitivity with respect to the time step.

3.2.2. Effect of Crack Tip Enrichment in Problems of Crack Propagation

Here, we demonstrate the necessity of incorporating enriched functions into the approximate solutions. A full model of the symmetric geometry and boundary conditions, as described in the previous section, was considered and is shown in Figure 13(a). The material behavior was assumed to be linear elastic. An initial crack, 5 mm in length, was placed on the middle left side of the plate. This crack was represented using XFEM, with the crack defined by the 1D cutting mesh, as described in Bajpai et al. [33]. The crack was prescribed to grow incrementally each step; the increment of $\Delta a = 0.17$ mm was chosen because it ensures that the crack tip extends arbitrarily within the computational domain. Nodal solutions in the elements surrounding the crack tip were enriched by the second term in Equation (11), consistent with the XFEM framework. Figures 13(b) and 13(c) illustrate the crack evolution over time, showing the crack propagating horizontally from the initial tip toward the opposite edge, as expected.



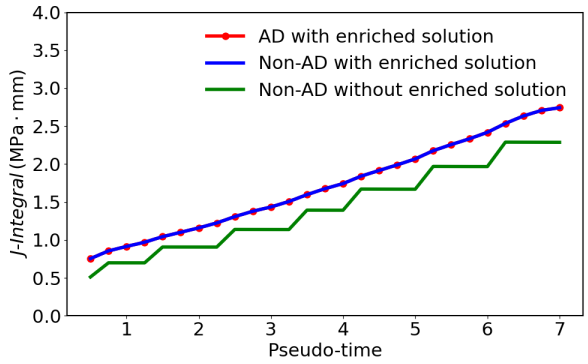
(a) (b) (c)

Figure 13. Tensile test model: (a) mesh configuration with an embedded crack on the left side of the plate, and (b–c) crack evolution over time visualized through the displacement field in the y-direction.

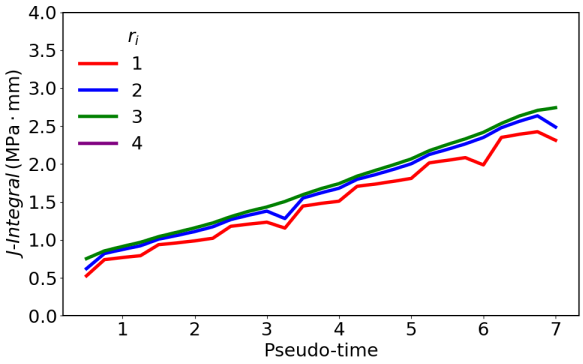
Figure 14(a) presents the time history of the J-integral for different implementations. First, we examine the effect of incorporating enriched functions. Both implementations that include enriched functions–regardless of whether AD was used–produced identical results, showing a smooth and continuous evolution of the J-integral. In contrast, the non-enriched implementation exhibits a step-ladder pattern, indicating that the crack tip is constrained to element boundaries. This highlights the importance of enriched functions, which allow the crack tip to be positioned arbitrarily within elements, thereby improving the accuracy and smoothness of the computed QoIs.

Next, we consider the role of AD. When enriched functions are included, both the AD and non-AD implementations yield consistent results, demonstrating that AD does not affect the accuracy in this case. However, AD offers advantages in terms of implementation simplicity and robustness, particularly when dealing with complex constitutive models.

The effect of the enriched domain, defined by the radius r_i, on the J-integral is shown in Figure 14(b).



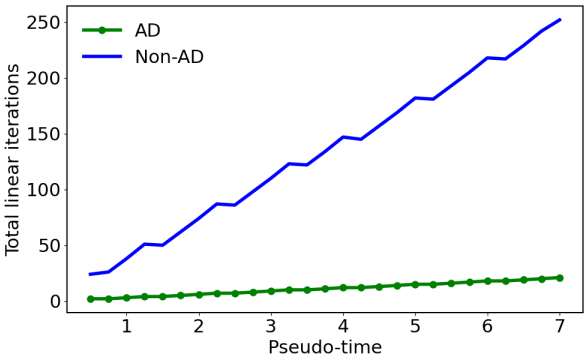
(a) Comparison between AD and non-AD implementations with and without enriched functions.



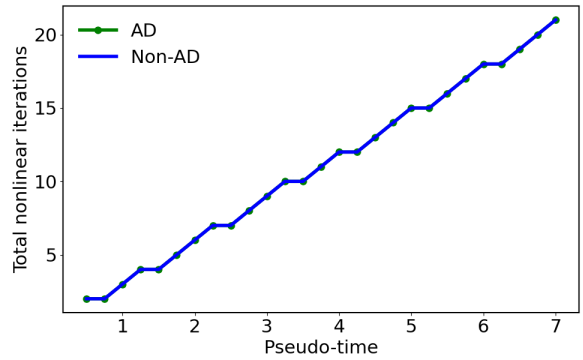
(b) Comparison across different contour radii.

Figure 14. Comparisons of computed values of the J-integral for the tensile test with various modeling options.

When r_i is set to one to two times the element size $h = 1$, small fluctuations in the J-integral persist. However, as r_i increases to at least three times h, these fluctuations disappear. Therefore, it is recommended to choose $r_i = 3h$ in practice. Finally, we demonstrate the efficiency of the AD implementation compared to the existing non-AD version. Figure 15(a) shows the total number of linear iterations for both approaches. In the AD implementation, the number of iterations increases linearly over time, reaching 24 at the final time, $t = 7$. In contrast, the non-AD requires significantly more linear iterations at each load step, resulting in a total of 252 iterations—10 times greater than the AD counterpart. On the other hand, as shown in Figure 15(b), the total number of nonlinear iterations remains the same for both implementations in this linear elastic problem.



(a) Total linear iterations.



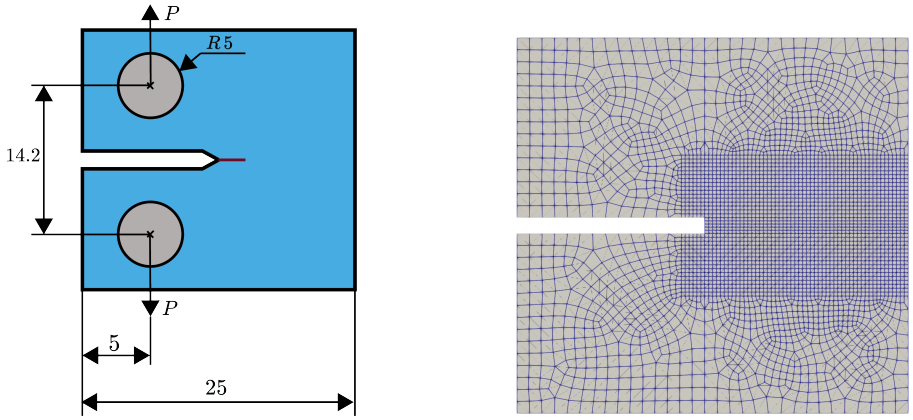
(b) Total nonlinear iterations.

Figure 15. Tensile test. Comparison of the performance of AD and non-AD implementations.

3.2.3. Validation

In this section, a compact-tension test specimen was used to evaluate the the capability of the developed XFEM to simulate crack growth under inelastic material behavior. The geometry and boundary conditions are illustrated in Figure 16(a). The specimen was subjected to a tensile load of $P = 205\,\mathrm{N}$ applied at the upper and lower pin holes. The mesh discretization, shown in Figure 16(b), features a finer resolution in the region where crack propagation was expected.

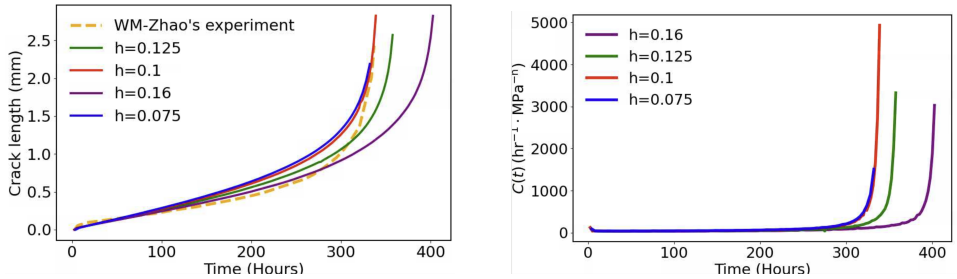
A study on mesh convergence for both the crack length and the contour integral was conducted. The



(a) Geometry and boundary conditions. (b) Mesh configuration.

Figure 16. Problem setup of the compact tension test. All dimensions are in mm.

results for crack length are shown in Figure 17(a), with mesh sizes of 0.16, 0.125, 0.1, and 0.075 mm. As the mesh is refined, the results converge at $h = 0.1$ mm. The material properties in Equation (28), which governs the crack growth rate, were calibrated as $B = 9 \times 10^{-4}$ and $m = 0.9$. These calibrated values show good agreement with experimental data for weld material in Zhao et al. [35].



(a) Accumulated crack length. (b) Contour integral C(t).

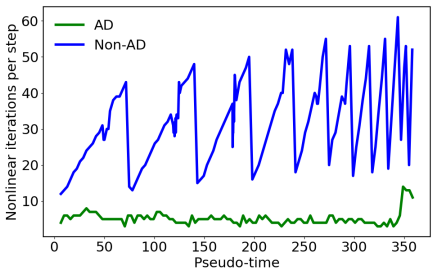
Figure 17. Compact tension test. Convergence study for crack length and contour integral $C(t)$.

Figure 17(b) presents the corresponding time history of the contour integral $C(t)$, which exhibits three distinct creep stages: primary, secondary, and tertiary. During primary creep, $C(t)$ decreases, followed by a plateau at a constant value C^* during secondary creep, spanning from $t = 5$ to 270 hours. In the tertiary stage, $C(t)$ increases sharply, leading to crack extension, as is also observed in Figure 17(a).

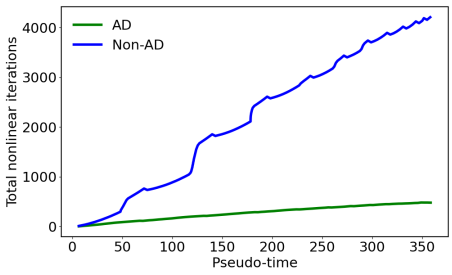
Next, we demonstrate the robustness of the AD implementation compared to the non-AD version. Figure 18(a) shows the number of nonlinear iterations per step. On average, the AD implementation requires about four iterations per step, while the non-AD version fluctuates at around 32. Figure 18(b) presents the accumulated number of nonlinear iterations over time; the AD implementation reached 480 and the non-AD version 4,280 at the final time of $t = 360$.

In summary, the AD implementation significantly outperforms the non-AD version in terms of nonlinear iteration efficiency. However, based on our numerical experience with other problems, if an exact Jacobian is implemented, the non-AD version can be approximately twice as fast as the AD counterpart.

Finally, the importance of crack tip enrichment is demonstrated in Figure 19. In Figure 19(a), the time history of the contour integral $C(t)$ exhibits noticeable oscillations in the case without enrichment (blue),



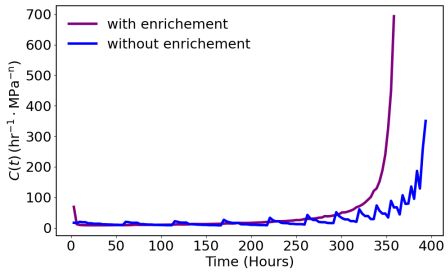
(a) Nonlinear iterations per time step.



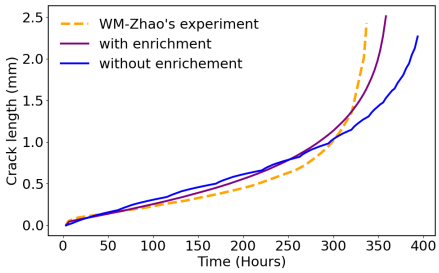
(b) Total nonlinear iterations.

Figure 18. Comparison of total nonlinear iterations between the AD and non-AD implementation for the compact tension test simulation.

whereas a smooth evolution is observed when enrichment is applied (purple). The latter behavior is desirable, as it reduces the sensitivity of $C(t)$ to mesh size. The corresponding crack lengths for both cases are shown in Figure 19(b). Due to the fluctuations in $C(t)$, the unenriched solution results in visible kinks in the crack length evolution, highlighting the necessity of enrichment for accurate and stable crack growth simulation.



(a) Contour integral C(t).



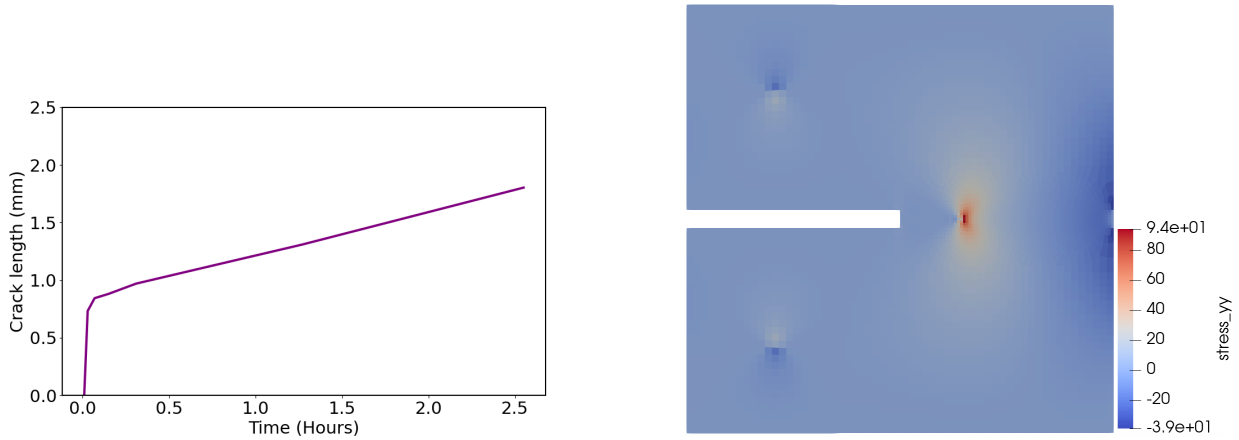
(b) Accumulated crack length.

Figure 19. Comparison of results with and without enriched solutions for the compact tension test.

3.2.4. Application of Surrogate Model for Simulating Crack Growth

Here we demonstrate the possibility of incorporating the MoE model for creep behavior, as described in Section 2, into the XFEM framework (Section 3). To compute the $C(t)$ integral using the MoE, where an explicit relationship between stress and strain is not provided, we used the incremental evaluation of SERD, as presented in Section 3.1.4.1. The same problem setup presented in the previous section (compact tension test) was used. Since the valid stress space in the MoE model exceeds the stress level resulting from the applied load $P = 205\,\mathrm{N}$, a reduced load of $P = 30\,\mathrm{N}$ was used in this example, with the material properties in Equation (28) chosen as $B = 10$ and $m = 0.9$.

Figure 20(a) shows the time history of the accumulated crack length up to the point where the numerical scheme fails in the secondary creep regime. The corresponding stress state in the y-direction at the failure time is visualized in Figure 20(b), highlighting a stress concentration of 94 MPa near the crack tip.



(a) Accumulated crack length.

(b) Contour plot of stress in the y-direction.

Figure 20. Results of compact tension test simulation using MoE model integrated into the MOOSE with crack growth represented using XFEM.

3.3. Summary and Future Work

In this section, we presented our recent developments in modeling CCG using XFEM. Numerical results were compared against analytical solutions and experimental data and demonstrated the capabilities of the proposed fracture model. In particular, including enrichment functions in the regions around crack tips enabled cracks to be placed arbitrarily within elements, making the method suitable for problems where crack extension is driven by a fracture criterion such as the $C(t)$ integral.

The AD approach outperformed the non-AD version in terms of reducing the number of nonlinear iterations. However, our numerical experience indicates that the non-AD implementation was two to three times faster than the AD-based one.

To model CCG, the Paris law-type relation was employed to link crack increment to the $C(t)$ integral. This required two additional calibration parameters. While we demonstrated the feasibility of incorporating a MoE surrogate model into the XFEM framework, further validation against realistic data remains necessary. Importantly, due to limitations in the valid stress space defined by the surrogate model, the XFEM code failed when the applied load resulted in stress levels exceeding this limit. Major areas for future work in the next stages of this project include calibrating the parameters for the CCG model against experimental data, improving the robustness of the surrogate model for XFEM simulations, and applying the methods developed here to Alloy 617.

4. REFERENCES

[1] N. Bieberdorf, A. Tallman, M. A. Kumar, V. Taupin, R. A. Lebensohn, and L. Capolungo, "A mechanistic model for creep lifetime of ferritic steels: Application to grade 91," *International Journal of Plasticity*, vol. 147, p. 103086, 2021.

[2] W. Wen, L. Capolungo, A. Patra, and C. Tomé, "A physics-based crystallographic modeling framework for describing the thermal creep behavior of Fe-Cr alloys," *Metallurgical and Materials Transactions A*, vol. 48, no. 5, pp. 2603–2617, 2017.

[3] D. Wang, C. Liu, J. Zhang, and L. Lou, "Evolution of grain boundary precipitates in a directionally solidified Ni-base superalloy during high temperature creep," *matrix*, vol. 6, p. 23, 2012.

[4] W. Wen, A. Kohnert, M. A. Kumar, L. Capolungo, and C. Tomé, "Mechanism-based modeling of thermal and irradiation creep behavior: An application to ferritic/martensitic HT9 steel," *International Journal of Plasticity*, vol. 126, p. 102633, 11 2019.

[5] J. Larsson, "Evaluation of current methods for creep analysis and impression creep testing of power plant steels," 2012.

[6] M. C. Egeberg, "Optimal design of validation experiments for calibration and validation of complex numerical models," Master's thesis, Clemson University, 2014.

[7] A. E. Tallman, M. Arul 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, no. 1, pp. 126–137, 2021.

[8] W. Wen, A. Kohnert, M. A. Kumar, L. Capolungo, and C. N. Tomé, "Mechanism-based modeling of thermal and irradiation creep behavior: An application to ferritic/martensitic HT9 steel," *International Journal of Plasticity*, vol. 126, p. 102633, 2020.

[9] F. Xue, T.-L. Cheng, Y. Lei, and Y.-H. Wen, "Phase-field modeling of crack growth under coupled creep-fatigue," *International Journal of Fatigue*, vol. 189, 8 2024.

[10] S. Wu, H. Song, H. Peng, P. Hodgson, H. Wang, X. Wu, Y. Zhu, M. Lam, and A. Huang, "A microstructure-based creep model for additively manufactured nickel-based superalloys," *Acta Materialia*, vol. 224, p. 117528, 2022.

[11] W. Wang, P. Buhl, A. Klenk, and Y. Liu, "A continuum damage mechanics-based viscoplastic model of adapted complexity for high-temperature creep–fatigue loading," *Journal of Engineering for Gas Turbines and Power*, vol. 138, no. 9, p. 092501, 2016.

[12] K. M. Z. Nasir, W. Jiang, and B. Spencer, "Creep and Creep Fracture Modeling with Surrogate Creep Models and the Extended Finite Element Method," Tech. Rep. INL/RPT-24-82173-Rev000, Idaho National Laboratory, Idaho Falls, ID (United States), 9 2024.

[13] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton, "Adaptive mixtures of local experts," *Neural computation*, vol. 3, pp. 79–87, 2 1991.

[14] P. Robbea, A. Ruybalid, A. Hegde, C. Bonneville, H. N. Najm, L. Capolungo, and C. Safta, "A comparison of surrogate constitutive models for viscoplastic creep: Simulation of HT-9 steel," *Under Review*, 2025. Manuscript under review.

[15] R. Lebensohn, C. Tomé, and P. Maudlin, "A selfconsistent formulation for the prediction of the anisotropic behavior of viscoplastic polycrystals with voids," *Journal of the Mechanics and Physics of Solids*, vol. 52, pp. 249–278, 9 2003.

[16] L. Harbour, G. Giudicelli, A. D. Lindsay, P. German, J. Hansel, C. Icenhour, M. Li, J. M. Miller, R. H. Stogner, P. Behne, D. Yankura, Z. M. Prince, C. DeChant, D. Schwen, B. W. Spencer, M. Tano, N. Choi, Y. Wang, M. Nezdyur, Y. Miao, T. Hu, S. Kumar, C. Matthews, B. Langley, N. Nobre, A. Blair, C. MacMackin, H. B. Rocha, E. Palmer, J. Carter, J. Meier, A. E. Slaughter, D. Andrš, R. W. Carlsen, F. Kong, D. R. Gaston, and C. J. Permann, "4.0 MOOSE: Enabling massively parallel multiphysics simulation," *SoftwareX*, vol. 31, p. 102264, 2025.

[17] F. P. Dunne and N. Petrinić, *Introduction to computational plasticity*. Oxford University Press eBooks, 6 2005.

[18] W. S. Slaughter, *The linearized theory of elasticity*. Birkhäuser Boston, MA, 1 2002.

[19] 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, Idaho Falls, ID (United States), Sept. 2022.

[20] 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.

[21] A. C. Hindmarsh, "Odepack, a systematized collection of ode solvers," in *Scientific Computing* (R. S. Stepleman *et al.*, eds.), vol. 1 of *IMACS Transactions on Scientific Computation*, pp. 55–64, Amsterdam: North-Holland, 1983.

[22] L. Petzold, "Automatic selection of methods for solving stiff and nonstiff systems of ordinary differential equations," *SIAM Journal on Scientific and Statistical Computing*, vol. 4, no. 1, pp. 136–148, 1983.

[23] 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.

[24] N. Sukumar, N. Moës, B. Moran, and T. Belytschko, "Extended finite element method for three-dimensional crack modelling," *International journal for numerical methods in engineering*, vol. 48, no. 11, pp. 1549–1570, 2000.

[25] C. L. Richardson, J. Hegemann, E. Sifakis, J. Hellrung, and J. M. Teran, "An XFEM method for modeling geometrically elaborate crack propagation in brittle materials," *International Journal for Numerical Methods in Engineering*, vol. 88, no. 10, pp. 1042–1065, 2011.

[26] A. Lindsay, R. Stogner, D. Gaston, D. Schwen, C. Matthews, W. Jiang, L. K. Aagesen, R. Carlsen, F. Kong, A. Slaughter, *et al.*, "Automatic differentiation in metaphysicl and its applications in moose," *Nuclear Technology*, vol. 207, no. 7, pp. 905–922, 2021.

[27] 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.

[28] H. Riedel, *Fracture at high temperatures*. Springer, 2014.

[29] K. Nikbin, "The fracture mechanics concepts of creep and creep/fatigue crack growth," in *Component Reliability under Creep-Fatigue Conditions*, pp. 177–242, Springer, 1998.

[30] Dassault Systèmes, *Abaqus 2024 Documentation*. SIMULIA, Dassault Systèmes, 2024. Available at: https://help.3ds.com/.

[31] M. Kumar and I. V. Singh, "Numerical investigation of creep crack growth in plastically graded materials using c (t) and xfem," *Engineering Fracture Mechanics*, vol. 226, p. 106820, 2020.

[32] T. L. Anderson and T. L. Anderson, *Fracture mechanics: fundamentals and applications*. CRC press, 2005.

[33] P. Bajpai, V. Prithivirajan, L. B. Munday, G. Singh, and B. W. Spencer, "Development of graphite thermal and mechanical modeling capabilities in Grizzly," Tech. Rep. INL/RPT-24-78905, Idaho National Laboratory, Idaho Falls, ID, May 2024.

[34] E. Hinton and M. Ezatt, *Fundamental tests for two & three dimensional, small strain, elastoplastic finite element analysis*. NAFEMS, 1987.

[35] L. Zhao, H. Jing, Y. Han, J. Xiu, and L. Xu, "Prediction of creep crack growth behavior in ASME P92 steel welded joint," *Computational materials science*, vol. 61, pp. 185–193, 2012.