DISCLAIMER

Argonne NATIONAL LABORATORY

ANL-25/23

Improved Weld Residual Stress Modeling System in BlackBear

Applied Materials Division

About Argonne National Laboratory
Argonne is a U.S. Department of Energy laboratory managed by UChicago Argonne, LLC under contract DE-AC02-06CH11357. The Laboratory's main facility is outside Chicago, at 9700 South Cass Avenue, Argonne, Illinois 60439. For information about Argonne and its pioneering science and technology programs, see www.anl.gov.

ANL-25/23

Improved Weld Residual Stress Modeling System in BlackBear

---

prepared by
Cheng-Hau Yang, Argonne National Laboratory
Mark C. Messner, Argonne National Laboratory
Tianchen Hu, Argonne National Laboratory

Applied Materials Division

August 2025

ERRATA

After initial publication of this report the authors were made aware of some minor errors. These errors were corrected in the present version on Sep 22, 2025.

Location	Error	Correction
Page 1	In the introduction, the second sentence in the third paragraph was imprecise and could lead to some potential confusion about the motivation of the current development.	The second sentence is reworded to “Implementing these advanced capabilities within an open-source framework such as the Multiphysics Object-Oriented Simulation Environment (MOOSE) allows user access to low-level code needed for specialized model development or large-scale parallel computing, providing a flexible, transparent, and highly scalable platform for developing custom multiphysics simulations.”

[This page left intentionally blank]

Abstract

This report presents enhancements to the MOOSE-based BlackBear application aimed at improving its capability to simulate welding and other thermo-mechanical manufacturing processes. Two primary avenues of improvement are pursued. First, to enhance user accessibility, we introduce a centralized default block restriction mechanism that ensures coverage checks are performed within user-specified default blocks. This default setting is applied consistently to all block-describable objects, such as variables, kernels, and more. In addition, we develop a modular action for moving heat source simulations, which integrates path file parsing, subdomain modification, and heat source kernel enforcement into a single, streamlined configuration. Second, to improve solver robustness, we implement an alternative method for assigning initial conditions to the updated active domain during the simulation, thereby enhancing convergence behavior. To validate the framework, we design and conduct several benchmark simulations, including heat conduction with progressive material addition, linear elasticity with time-dependent material deposition, and viscoplasticity model with isotropic hardening under similar conditions. Finally, we demonstrate the effectiveness of the proposed framework through large-scale thermo-mechanical welding simulations in both two and three dimensions.

Table of Contents

List of Figures

List of Tables

1 Introduction

Welding is a widely used manufacturing and repair process across industries ranging from power generation to transportation. The process involves the application of a highly concentrated heat source, creating steep thermal gradients in the welded component. These gradients cause non-uniform plastic deformation — commonly referred to as welding-induced plasticity — which in turn generates residual stresses and distortions in the structure. Such residual stresses can significantly influence the structural integrity and service life of engineering components by superimposing on operational loads, potentially exceeding design limits. Similarly, welding-induced distortions can adversely affect dimensional tolerances, assembly processes, and overall product performance.

Direct experimental measurement of residual stresses, while valuable, is often time-consuming, costly, and in many cases impractical — especially for large or complex components. Moreover, relying solely on physical trials to optimize welding parameters for residual stress mitigation can be prohibitively expensive. Consequently, computational welding mechanics has emerged as a powerful tool for predicting thermal fields, plasticity, residual stresses, and distortions, enabling informed decision-making in process design and component assessment.

Although numerous thermo-mechanical welding models exist, many are implemented within commercial finite element software such as ABAQUS, ANSYS, or SYSWELD. Implementing these advanced capabilities within an open-source framework such as the Multiphysics Object-Oriented Simulation Environment (MOOSE) [2] allows user access to low-level code needed for specialized model development or large-scale parallel computing, providing a flexible, transparent, and highly scalable platform for developing custom multiphysics simulations.

BlackBear, a MOOSE-based application for solid mechanics, extends these capabilities by integrating advanced constitutive models from the Nuclear Engineering Material model Library (NEML) and its offshoot NEML2 [1]. This combination offers a versatile foundation for welding and additive manufacturing simulations. However, modeling complex manufacturing processes such as multi-pass welding in MOOSE presents challenges, particularly in domain activation, variable (re)initialization, and efficient heat source modeling.

This report describes recent developments aimed at improving the robustness, usability, and computational efficiency of welding simulations within the BlackBear application. Key contributions include a centralized block restriction mechanism for simplifying domain-specific simulation setup, a modular action for moving heat source simulations, and enhanced algorithms for initializing solution variables in newly activated domains. These capabilities are demonstrated through verification problems and large-scale thermo-mechanical simulations, showcasing their potential for advancing open-source welding simulation technology.

2 Mathematical formulation

2.1 Energy balance

With a view towards verification, we consider both the steady-state energy balance and its transient counterpart. The steady-state heat conduction equation is written as

$$\nabla \cdot (k \nabla T) = 0, \tag{2.1}$$

with appropriate boundary conditions and negligible heat source/sink. Here, T is the temperature, and k is the thermal conductivity.

The transient heat conduction equation is given by

$$\rho C_p \dot{T} = \nabla \cdot (k \nabla T), \tag{2.2}$$

with additional appropriate initial conditions. Here, ρ is the density, C_p is the specific heat capacity.

2.2 Linear momentum balance

Thermo-mechanical coupling is of particular interest when modeling additive manufacturing processes as its the primary source of residual stress. For verification purposes, we consider the balance of linear momentum under small strain assumptions, i.e.

$$\nabla \cdot \boldsymbol{\sigma} = 0, \tag{2.3}$$

with appropriate boundary conditions and negligible body force, where $\boldsymbol{\sigma}$ is the Cauchy stress.

Two constitutive models are considered for verification purposes. The linear isotropic elasticity model serves as the basis of verification free of any history-dependent data. A rate-dependent viscoplasticity model is used to examine the implementation of stateful internal variable management.

For linear isotropic elasticity, the constitutive relation is given as

$$\boldsymbol{\sigma} = K \operatorname{tr}(\boldsymbol{\varepsilon}) + 2G \operatorname{dev}(\boldsymbol{\varepsilon}), \tag{2.4}$$

where K and G are bulk and shear moduli, respectively.

The rate-dependent viscoplasticity model with isotropic hardening under consideration is

written as:

$$\boldsymbol{\varepsilon} = \boldsymbol{\varepsilon}^e + \boldsymbol{\varepsilon}^p, \tag{2.5a}$$

$$\boldsymbol{\sigma} = K \operatorname{tr}(\boldsymbol{\varepsilon}^e) + 2G\operatorname{dev}(\boldsymbol{\varepsilon}^e), \tag{2.5b}$$

$$f = \bar{\sigma} - \sigma_y - k(\bar{\varepsilon}^p), \tag{2.5c}$$

$$\dot{\gamma} = \left(\frac{\langle f \rangle}{\eta}\right)^n, \tag{2.5d}$$

$$\dot{\boldsymbol{\varepsilon}}^p = \dot{\gamma}\frac{\partial f}{\partial \boldsymbol{\sigma}}, \tag{2.5e}$$

Here, $\boldsymbol{\varepsilon}^e$ and $\boldsymbol{\varepsilon}^p$ denote the elastic and plastic strains, respectively. The equivalent plastic strain $\bar{\varepsilon}^p$ is a scalar internal variable that accumulates plastic deformation. An associative J_2 plastic flow is used together with isotropic hardening. The evolution of internal variables is integrated implicitly in time using the backward Euler scheme.

Both constitutive models are implemented in the NEML2 framework [1]. An example input file for the viscoplasticity model is shown below.

```
[Models]
  [mandel_stress]
    type = IsotropicMandelStress
  []
  [vonmises]
    type = SR2Invariant
    invariant_type = 'VONMISES'
    tensor = 'state/internal/M'
    invariant = 'state/internal/s'
  []
  [isoharden]
    type = LinearIsotropicHardening
    hardening_modulus = 1000
  []
  [yield]
    type = YieldFunction
    yield_stress = 5
    isotropic_hardening = 'state/internal/k'
  []
  [flow]
    type = ComposedModel
    models = 'vonmises yield'
  []
  [normality]
    type = Normality
    model = 'flow'
    function = 'state/internal/fp'
```

```
  from = 'state/internal/M state/internal/k'
  to = 'state/internal/NM state/internal/Nk'
[]
[flow_rate]
  type = PerzynaPlasticFlowRate
  reference_stress = 100
  exponent = 2
[]
[Eprate]
  type = AssociativePlasticFlow
[]
[eprate]
  type = AssociativeIsotropicPlasticHardening
[]
[Erate]
  type = SR2VariableRate
  variable = 'forces/E'
[]
[Eerate]
  type = SR2LinearCombination
  from_var = 'forces/E_rate state/internal/Ep_rate'
  to_var = 'state/internal/Ee_rate'
  coefficients = '1 -1'
[]
[elasticity]
  type = LinearIsotropicElasticity
  coefficients = '1e5 0.3'
  coefficient_types = 'YOUNGS_MODULUS POISSONS_RATIO'
  rate_form = true
[]
[integrate_stress]
  type = SR2BackwardEulerTimeIntegration
  variable = 'state/S'
[]
[integrate_ep]
  type = ScalarBackwardEulerTimeIntegration
  variable = 'state/internal/ep'
[]
[implicit_rate]
  type = ComposedModel
  models = 'mandel_stress vonmises isoharden yield normality
 flow_rate Eprate eprate Erate Eerate elasticity integrate_stress
 integrate_ep'
[]
[model]
```

```
    type = ImplicitUpdate
    implicit_model = 'implicit_rate'
    solver = 'newton'
  []
[]
```

3 Domain definition

Before presenting our welding simulation results and validation examples, we first define the domains relevant to our current study. The sketch in Figure 3.1 illustrates the domains considered in the subsequent analysis:

- The *active* domain where equations are defined.
- The *inactive* domain (■) where equations are *not* defined.
- The *stationary active* domain (■), a subset of the *active* domain whose subdomain ID has not been changed in the current domain update.
- The *updated active* domain (■), a subset of the *active* domain whose subdomain ID has changed in the current domain update.

The following requirements/assumptions apply to these domain definitions:

- Union of the *active* and the *inactive* domain is the entire domain.
- Intersection of the *active* and the *inactive* domain is the empty set $\emptyset$.
- The *stationary active* domain and the *updated active* domain are subsets of the *active* domain.
- Union of the *stationary active* domain and the *updated active* domain is the *active* domain.
- Intersection of the *stationary active* domain and the *updated active* domain is the empty set $\emptyset$.

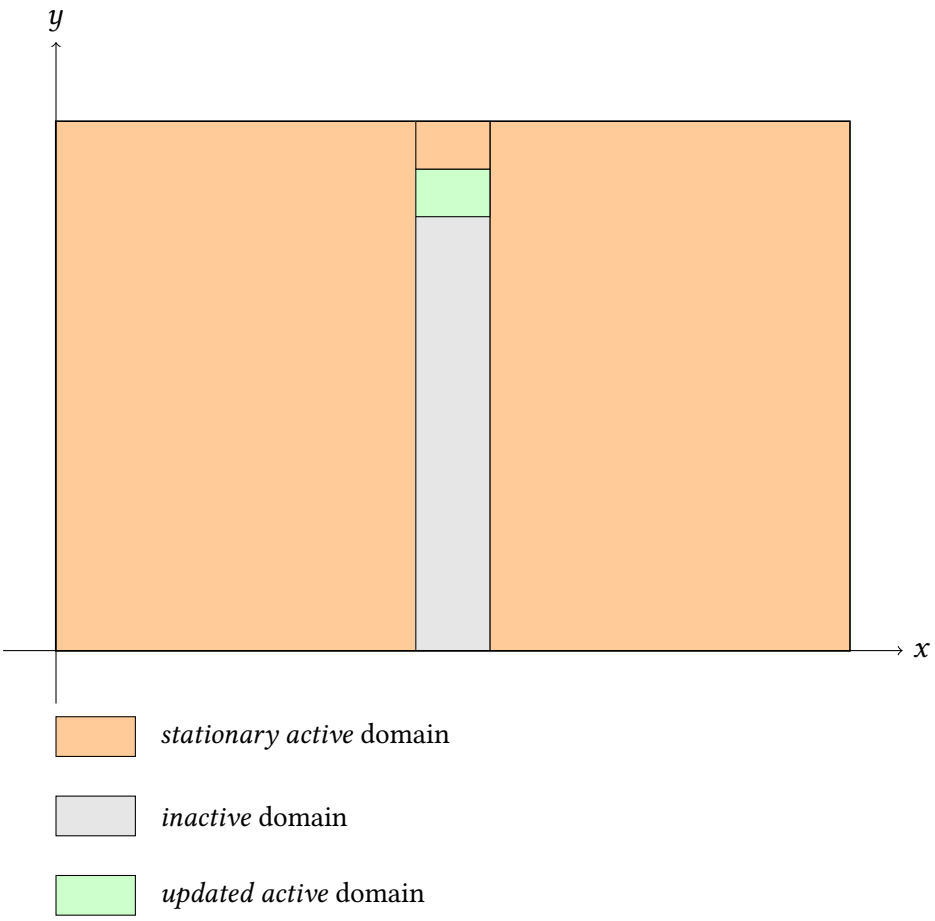


Figure 3.1: Overview of the computational domain.

4 Implementation

4.1 Variable (re)initialization on the *updated active* domain

In element activation simulations (e.g., for additive manufacturing), the degrees of freedom (DoFs) on the *updated active* domain have no prior solution information when they are incorporated into the solution process. In moving interface problems, the DoFs on the *updated active* domain already exist but may need to be re-initialized depending on the physics. As a result, it is necessary to selectively impose appropriate initial conditions for these DoFs.

There are two strategies for (re)initializing variables on the *updated active* domain. The first, arguably the simplest, strategy is to project the initial condition of the variable onto the *updated active* domain.

However, directly imposing such initial conditions often poses significant challenges for the solution procedure. For nonlinear material models, suboptimal initial guess could hinder convergence of the material model and/or that of the global solve. For evaluations on the deformed mesh, e.g., when geometric nonlinearity or mechanical contact is enabled, a bad initial condition of displacement could lead to ill-conditioned elements, and in extreme cases, could lead to highly skewed elements with inverted Jacobians.

The second strategy aims to address these challenges. The key idea is to initialize the solution on the *updated active* domain leveraging the solution information from the *stationary active* domain. In the current implementation, we adopt the Zienkiewicz-Zhu patch recovery technique, as implemented in `MOOSE` [2], to initialize the solution field for the DoFs on *updated active* elements.

The initialization algorithm starts with constructing a patch of elements from the *stationary active* domain. By definition, solution to the variable of interest already exists on the patch. The solution is first interpolated onto the quadrature points on the *stationary active* domain, and a polynomial is fitted against the interpolated variable values using a least-squares fit. A complete set of monomials $P_{\boldsymbol{\alpha}}$ up to a given order p is used as the basis of the polynomial. Let $\boldsymbol{\alpha}$ be a multi-index of dimension d, where

$$\boldsymbol{\alpha} = (\alpha_1, ..., \alpha_d), \quad \alpha_i \in \mathbb{N}_{\geq 0}, \quad |\boldsymbol{\alpha}| = \sum_{i=1}^{d} \alpha_i. \tag{4.1}$$

The monomial basis for degree m can be written as

$$P_{|\boldsymbol{\alpha}|=m} = \{x^{\alpha_1} y^{\alpha_2} z^{\alpha_3}\}_{\boldsymbol{\alpha}}. \tag{4.2}$$

For example, in 2D ($d = 2$) with $p = 2$, the complete monomial basis can be written as $\boldsymbol{P} = [1, x, y, x^2, xy, y^2]$.

Remark. The monomial generation can be implemented recursively. Define $P(0) = C$ as the constant term. Higher-order monomials are then constructed by augmenting $P(n-1)$ with all combinations of $(\alpha_1, \alpha_2, \alpha_3)$ such that $\alpha_1 + \alpha_2 + \alpha_3 = n$. That is,

$$\boldsymbol{P}(n) = \boldsymbol{P}(n-1) \cup \{x^{\alpha_1} y^{\alpha_2} z^{\alpha_3} \mid \alpha_1 + \alpha_2 + \alpha_3 = n\}. \tag{4.3}$$

The fitting process solves a least-squares problem. In matrix form, it can be expressed as

$$\boldsymbol{A}\boldsymbol{c} = \boldsymbol{b}, \tag{4.4}$$

where $\boldsymbol{A}$ is the patch interpolation matrix, $\boldsymbol{b}$ is the patch interpolation vector, and $\boldsymbol{c}$ contains the fitted polynomial coefficients. These matrices are assembled over n quadrature points on the patch:

$$\begin{aligned} \boldsymbol{A} &= \sum_{i=1}^{n} \boldsymbol{P}^T(\boldsymbol{x}_i)\boldsymbol{P}(\boldsymbol{x}_i), \\ \boldsymbol{b} &= \sum_{i=1}^{n} \boldsymbol{P}^T(\boldsymbol{x}_i)u(\boldsymbol{x}_i), \end{aligned} \tag{4.5}$$

where $u(\boldsymbol{x}_i)$ is the interpolated variable value at the quadrature point $\boldsymbol{x}_i$.

Remark. For least-squares fitting with the same patch of elements, the computed coefficients $\boldsymbol{c}$ can be cached and reused to reduce computational overhead.

Once the least-squares problem is solved, the fitted polynomial is used to extrapolate the solution onto the *updated active* domain by projecting the polynomial.

4.2 Centralized block restriction for element subdomain modifiers

In simulations with evolving subdomains, such as those of additive manufacturing or welding processes, users often need to restrict the computations to a specific set of subdomains. However, in the current `MOOSE` framework, users are required to manually specify these block restrictions repeatedly in multiple locations within the input file — including the `block` parameter for each `Kernel`, `Material`, and `Variable`.

Moreover, as part of the current sanity checking step — where all elements are expected to be associated with an active `Kernel`, and `Material` properties must be provided for every element — users are also required to explicitly disable the `kernel_coverage_check` and `material_coverage_check` options when their simulations are restricted to a subset of the mesh. This process is error-prone, especially if users forgot to disable these checks or misconfigured block restrictions.

To address these limitations, we implemented a centralized mechanism for controlling block restrictions in `MOOSE`. With this new design, users only need to specify restricted subdomains using the `block` parameter under the `[GlobalParams]` section. Once specified, all coverage checks for `Kernel` and `Material` objects are automatically limited to the user-defined blocks. Furthermore, the block restriction is implicitly applied to all objects inheriting from the `BlockRestrictable` class, eliminating the need to repetitively define the same block restrictions throughout the input file.

This enhancement provides a streamlined framework for setting up simulations, particularly for welding and other additive manufacturing processes, where subdomain-restricted solving strategies are frequently employed.

For example, the following input file

```
[Problem]
  kernel_coverage_check = false
  material_coverage_check = false
[]

[Variables]
  [T]
    order = FIRST
    block = '0 1 3'
  []
[]

[Kernels]
  [diff]
    type = HeatConduction
    variable = T
    block = '0 1 3'
  []
[]
```

can now be simplified to

```
[GlobalParams]
    block = '0 1 3'
[]

[Variables]
  [T]
    order = FIRST
  []
[]

[Kernels]
  [diff]
    type = HeatConduction
    variable = T
  []
[]
```

where the variable `T` and and the kernel `diff` are block-restricted to subdomains `'1 2 3'`, and the kernel coverage check is automatically restricted to the same set of subdomains.

4.3 A Modular action for moving heat source simulation

The `SpatioTemporalHeatAction` is a custom `MOOSE Action` developed to streamline the setup of a spatio-temporal heat source. It automates the creation and coordination of four essential components for transient thermal simulations:

1. **UserObject: `CSVPiecewiseLinearSpatioTemporalPath`**
   Defines the time-dependent path of the moving heat source based on a CSV input file. This user object enables interpolation of the source location over time.
2. **MeshModifier: `SpatioTemporalPathElementSubdomainModifier`**
   Dynamically modifies the subdomain of elements within a specified distance from the moving path.
3. **Material: `ADMovingEllipsoidalHeatSource`**
   Provides an ellipsoidal volumetric heat source localized around the current position of the path.
4. **Kernel: `ADMatHeatSource`**
   Inserts the volumetric heat source term into the weak form of the governing heat equation, linking it to the temperature field variable.

By combining these components under one `Action`, we simplified user input by encapsulating weld path specification, mesh modification, source material response, and kernel integration in a single configuration interface.

Given the following configuration:

```
[SpatioTemporalHeat]
  path_file = 'concentric_circles_reverse.csv'
  ## for path
  verbose = true
  ## for esm
  block = '0 1'
  radius = 0.03
  execute_on_esm = 'TIMESTEP_BEGIN'
  old_subdomain_reinitialized = false
  reinitialize_subdomains = '0'
  reinitialization_strategy = "POLYNOMIAL_NEIGHBOR"
  reinitialize_variables = "T"
  polynomial_fitters = 'extrapolation_patch_T '
  ## for heat source
  power = 1
  a = 0.035
  b = 0.01
  efficiency = 1
  scale = 1
```

```
  ## for kernel
  heat_variable = cond
[]
```

The SpatioTemporalHeatAction action will automatically generate the equivalent set of objects shown below:

```
[SpatioTemporalPaths]
  [path]
    type = CSVPiecewiseLinearSpatioTemporalPath
    file = 'concentric_circles_reverse.csv'
    verbose = true
  []
[]

[MeshModifiers]
  [esm]
    type = SpatioTemporalPathElementSubdomainModifier
    path = 'path'
    radius = 0.03
    target_subdomain = '0'
    block = '0 1'
    execute_on = 'TIMESTEP_END'
    old_subdomain_reinitialized = false
    reinitialize_subdomains = '0'
    reinitialization_strategy = "POLYNOMIAL_NEIGHBOR"
    reinitialize_variables = "T"
    polynomial_fitters = 'extrapolation_patch_T '
  []
[]

[Materials]
  [volumetric_heat]
    type = ADMovingEllipsoidalHeatSource
    path = 'path'
    power = 1
    efficiency = 1
    scale = 1
    a = 0.035
    b = 0.01
  []
[]

[Kernels]
  [hsource]
```

```
    type = ADMatHeatSource
    material_property = 'volumetric_heat'
    variable = cond
  []
[]
```

5 Validation

To validate and demonstrate the new functionalities of the element subdomain modifier, we performed four validation problems as formulated in Section 2.

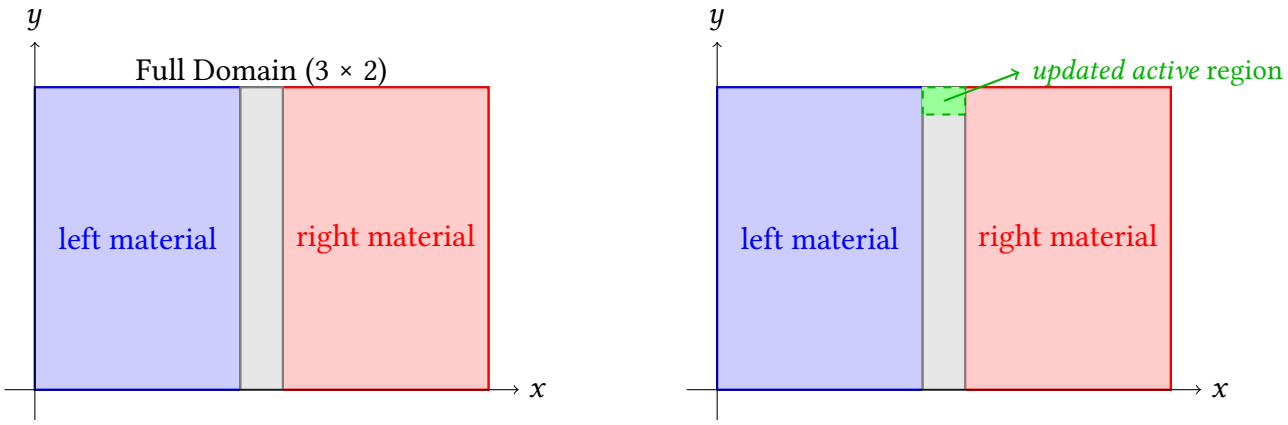
5.1 Steady-state heat conduction

Consider a domain $[0, 3] \times [0, 2]$ on which the steady-state heat conduction problem (i.e., (2.1)) is solved. Different material properties are assigned to each subdomain, as summarized in Table 5.1. Dirichlet boundary conditions are applied on the left and the right walls. Zero heat flux is assumed on all other boundaries, including those evolving with the changing subdomain.

A schematic of the computational domain is provided in Figure 5.1. The domain is divided into three regions: the left material domain, the right material region, and an initially *inactive* domain in between. Specifically, the left material domain spans from $x = 0$ to $x = 1.36$, the right material domain spans from $x = 1.64$ to $x = 3.0$, and the initially *inactive* domain lies between them, spanning from $x = 1.36$ to $x = 1.64$. The height of all three regions extends from $y = 0$ to $y = 2.0$. The *inactive* domain is initially excluded from the solution process. These elements are progressively activated during the simulation to mimic material deposition or weld progression. This setup enables us to evaluate the behavior of dynamic element activation.

The domain is discretized using linear quadrilateral elements with an element size of 0.02. A constant time step size of 1 is used throughout the simulation. The *inactive* domain is activated with a velocity of 0.06, starting from the top. In other words, the lower boundary of the *active* domain progresses following the relation $y = 2 - 0.06t$.

Solutions at four different times are verified against reference solutions where the corresponding *inactive* domain is predefined as *active* domain without the use of element subdomain modifier. It is observed that the L_2 error on the *active* domain is on the order of machine precision.



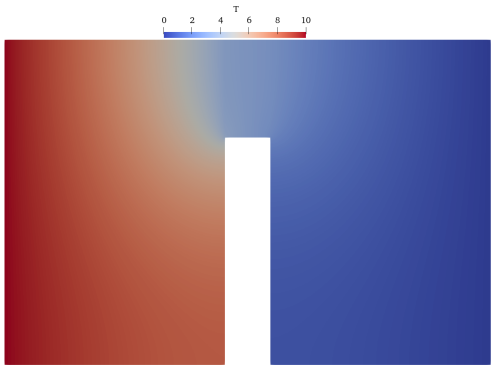
(a) Initial configuration.

(b) *updated active* region at the top.

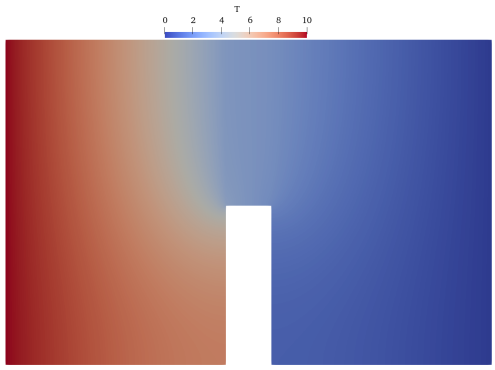
Figure 5.1: Schematic of the computational domain. (a) Initial configuration with three regions. (b) *updated active* region (in green) is activated progressively from the top of the *inactive* region.

Table 5.1: Material properties assigned to different blocks in the simulation.

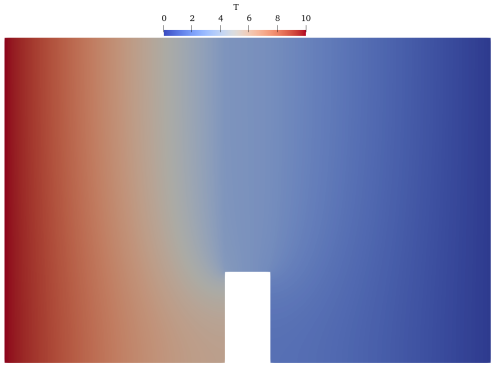
block	Specific Heat	Thermal Conductivity	Density
0 (Left)	30	20	10
1 (Right)	75	50	20
3 (Middle)	150	100	50



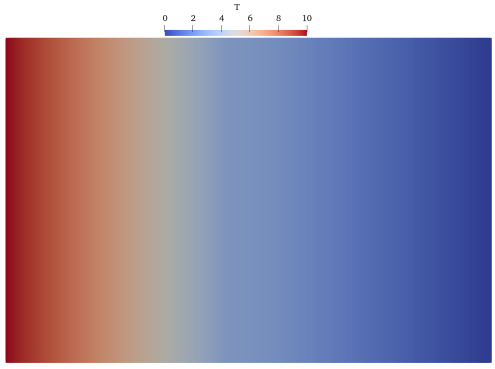
(a) $t = 10$ and hole height = 1.4



(b) $t = 17$ and hole height = 0.98



(c) $t = 24$ and hole height = 0.56



(d) $t = 34$ and no hole

Figure 5.2: Temperature contours at different times during the steady-state heat conduction simulation with dynamic element activation (Section 5.1).

5.2 Transient heat condition

After completing the validation in Section 5.1, we further incorporate the time derivative term into the strong form, describing a transient heat conduction problem (i.e. (2.2)). As the solution is history-dependent, this example serves to validate the implementation of solution state advancement. The computational domain follows the same configuration described in Section 5.1, with minor adjustments to the widths of individual subdomains to ensure consistency across mesh

refinement levels. Specifically, in this test case, the widths of the left and right material regions are both set to 1.25, while the width of the *inactive* region is set to 0.5.

With mesh refinement, the element size is denoted as 2^{-R}, with R being the refinement level ranging from 2 to 8. An activation speed of 2^{-4} is used. Note that for coarse meshes with $R = 2$ and $R = 3$, the element size is larger than 2^{-4}, and the *active* region is not being continuously activated at each time step, contributing to additional time discretization error.

Results of the mesh convergence study is shown in Figure 5.3. The average temperature on the *active* domain for $x \in [1.25, 1.75]$ is plotted as a function of time. The results demonstrate that the solution converges as the mesh is refined. The error remains reasonable as long as the spatial and temporal discretizations are able to resolve the activation speed, i.e. with $R \geq 4$.

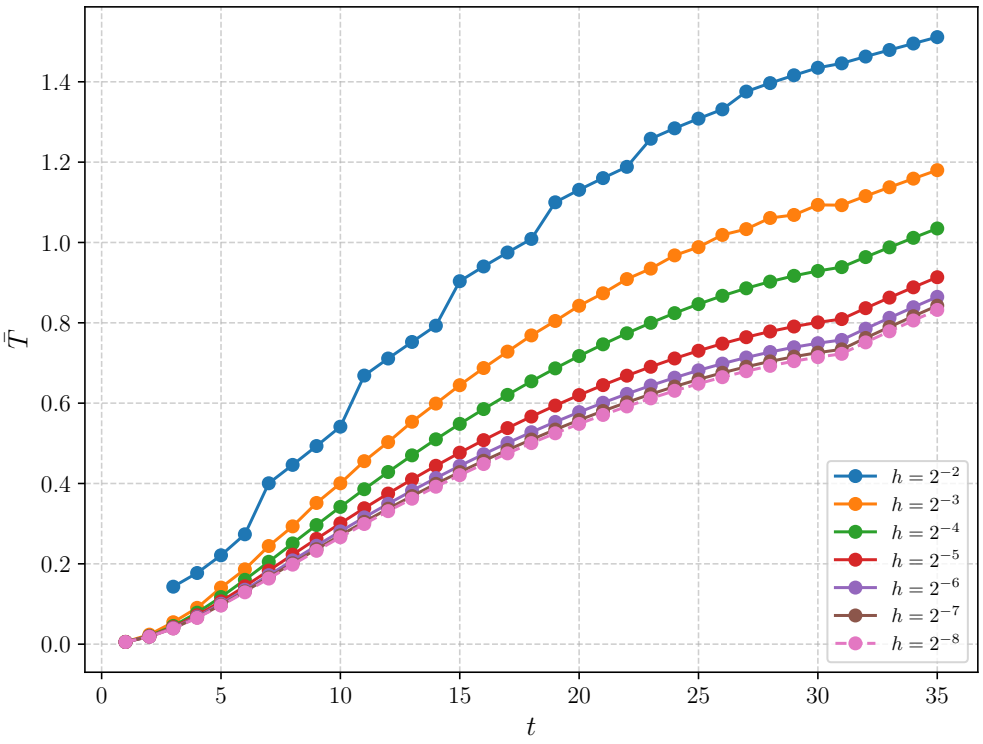


Figure 5.3: Mesh convergence study for the heat conduction problem with the time derivative term (Section 5.2). The mesh sizes are 2^{-R}, with R ranging from 2 to 8.

5.3 Path-independent constitutive model

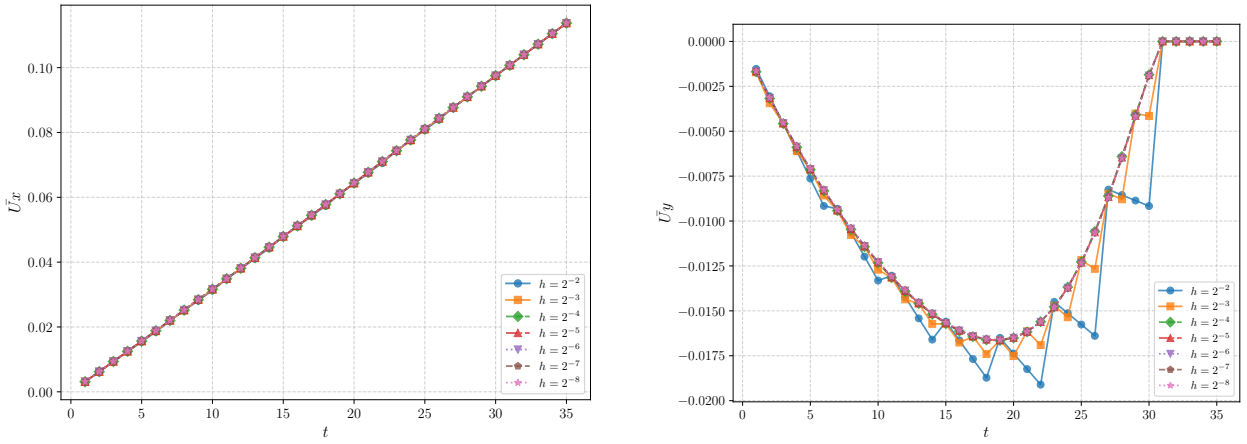
After validating the element subdomain modifier for the heat conduction problems in Section 5.1 and Section 5.2, we proceed to test the element subdomain modifier in mechanics simulations. In this section, we solve balance of linear momentum for an isotropic, elastic material under small strain assumptions, i.e., (2.4). The subdomain configurations and element activation strategy remain the same as those described in Section 5.2. The left wall is fully clamped; on the right wall, the y-displacement is fixed to zero and a displacement control is applied in the x-direction with

$u_x = 0.005t$; the boundaries are otherwise traction free, including those evolving with domain activation. Different material properties are assigned to different blocks as summarized in Table 5.2.

We again conduct a mesh convergence study by comparing the average displacements of the middle section. The convergence results for the average displacements are shown in Figure 5.4. In addition, the simulation results at different times with different mesh resolutions are visualized in Figure 5.5.

Table 5.2: Mechanics properties assigned to different blocks in the simulation.

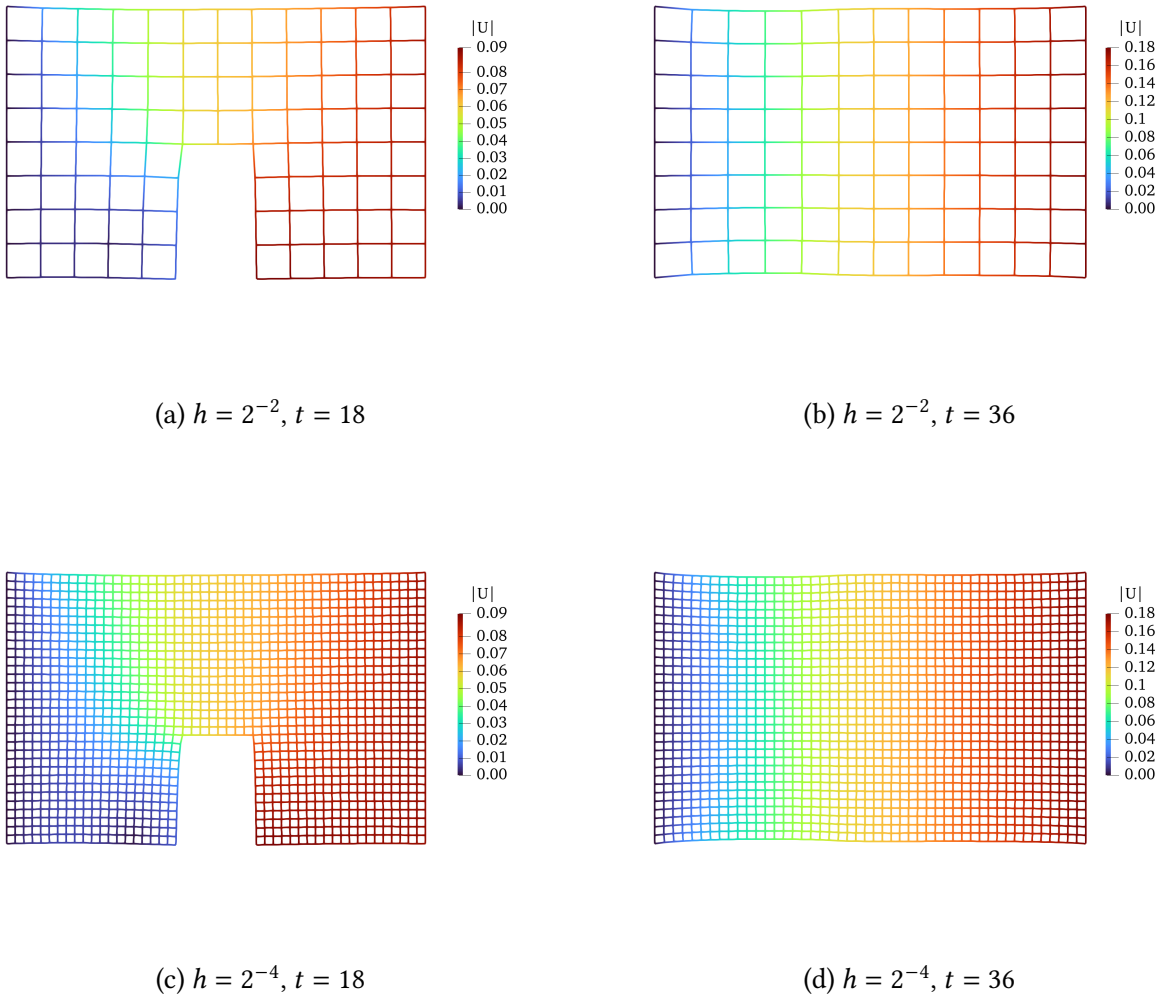
block	**Young's modulus**	**Poisson ratio**
0 (Left)	1	0.3
1 (Right)	2	0.3
3 (Middle)	3	0.3



(a) Averaged displacement in the x-direction

(b) Averaged displacement in the y-direction

Figure 5.4: Mesh convergence study of linear isotropic elasticity with the element subdomain modifier (Section 5.3). The average displacements in both x- and y-directions are shown at different mesh refinement levels.



(a) $h = 2^{-2}$, $t = 18$

(b) $h = 2^{-2}$, $t = 36$

(c) $h = 2^{-4}$, $t = 18$

(d) $h = 2^{-4}$, $t = 36$

Figure 5.5: Contour of the displacement magnitude of linear isotropic elasticity simulations (Section 5.3). The element size is $h = 2^{-2}$ and 2^{-4}, respectively. The results are presented at $t = 18$ and $t = 36$ for each case.

5.4 Path-dependent constitutive model

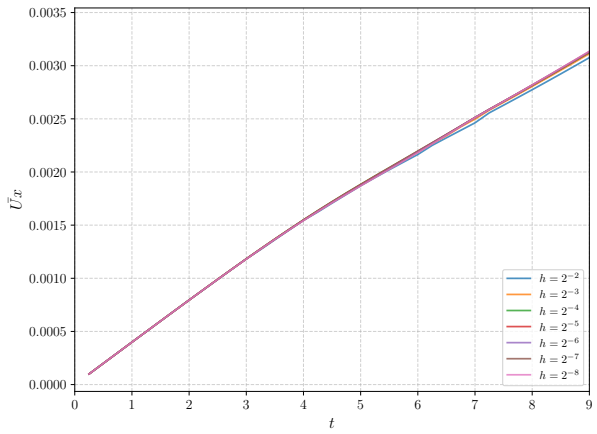
After completing the validation of linear isotropic elasticity in Section 5.3, we proceed to investigate the material models that are path-dependent. In this example, we employ a viscoplasticity model with isotropic hardening, i.e., (2.5). Table 5.3 summarizes the parameters of the material model.

A constant time step size of 0.01 is used. The subdomain configuration remains identical to those described in Section 5.2 and Section 5.3. However, the domain activation strategy employed here differs from the previous examples, where an initial *updated active* domain is activated at the top with a height of 2^{-2}. The boundary conditions are the same as Section 5.3 with $u_x = 0.0008t$.

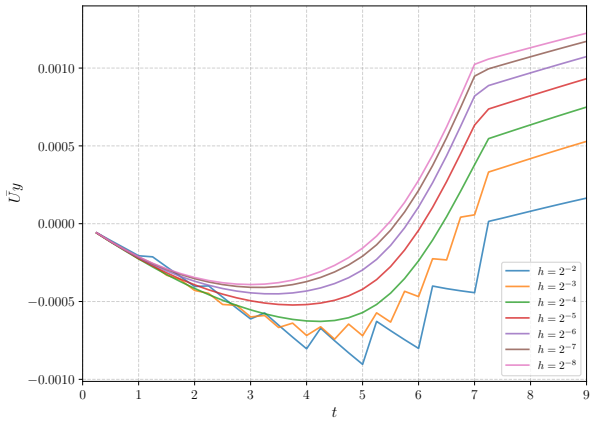
Table 5.3: Material parameters used in the viscoplasticity model with linear isotropic hardening

Symbol	Description	Value
E	Young's modulus	1×10^5
ν	Poisson's ratio	0.3
σ_y	Initial yield stress	5
H	Hardening modulus	1000
η	Perzyna reference stress	100
n	Perzyna exponent	2

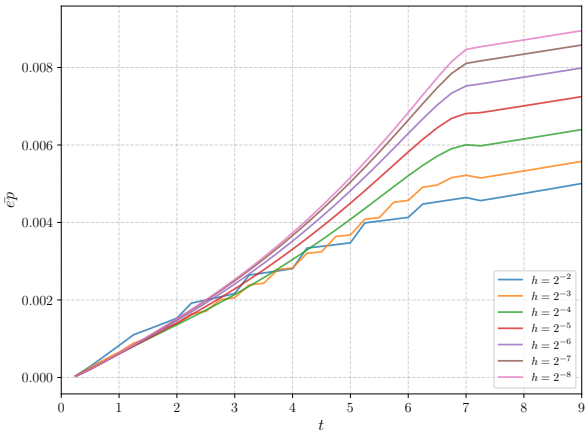
Both the average displacements and the average equivalent plastic strain of the middle section are examined in the mesh convergence study. Results are presented in Figure 5.6; a reasonable accuracy is attained as long as the spatial and temporal discretization resolves the activation speed. Mesh deformation at different times across different mesh refinement levels are illustrated in Figure 5.7.



(a) Average displacement in the x-direction.

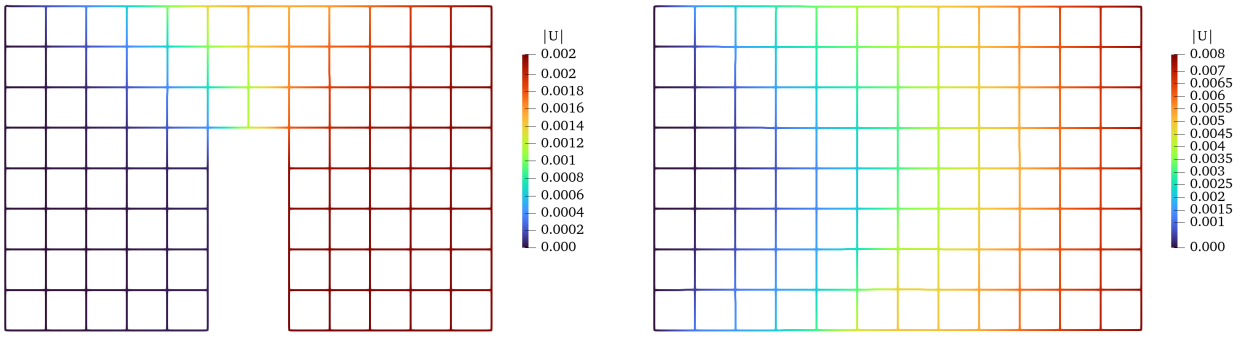


(b) Average displacement in the y-direction.



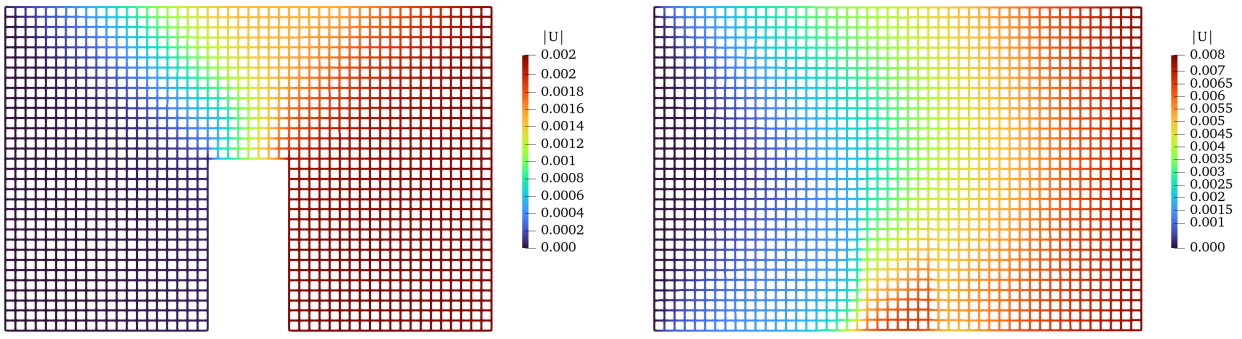
(c) Average equivalent plastic strain.

Figure 5.6: Mesh convergence study of the path-dependent viscoplasticity model (Section 5.4) with domain activation. The average displacements and equivalent plastic strain are shown for different mesh refinement levels.



(a) $h = 2^{-2}$, $t = 2.75$

(b) $h = 2^{-2}$, $t = 9$



(c) $h = 2^{-4}$, $t = 2.75$

(d) $h = 2^{-4}$, $t = 9$

Figure 5.7: Contour of displacement magnitude of the viscoplasticity simulations (Section 5.4). The element size is $h = 2^{-2}$ and 2^{-4}, respectively. The results are presented at $t = 2.75$ and $t = 9$ for each case.

6 Numerical examples

Two welding processes are simulated for demonstration purposes. Values used in the following examples are chosen solely for demonstration purposes and do not represent parameters of any real material.

6.1 Two-dimensional thermo-mechanical simulation of slot welding

A square plate of side length $L = 1$, with a circular hole of radius $R = 0.2L$, is discretized using linear quadrilateral elements (Figure 6.1a). The mesh is divided into *inactive* and *active* domains. The weld path is illustrated in Figure 6.1b. The path starts from the outermost circular layer and spirals inward in a continuous rotational pattern.

We solve a one-way coupled thermo-mechanical problem, where temperature evolution influences structural deformation, but not vice versa. The thermal behavior is governed by the transient heat conduction equation (2.2), with an added heat source term $q(\mathbf{x}, t)$.

The material parameters are as follows: thermal conductivity $k = 45$, specific heat $c_p = 0.5$, and density $\rho = 8000$. The ellipsoidal heat source $q(\mathbf{x}, t)$ has a Gaussian profile with semi-axes $a = 0.035\,L$ (in-plane) and $b = 0.01\,L$ (through-thickness), continuously reorienting with the weld path. The total power of the heat source is $P = 1$ with an efficiency factor $\eta = 1$.

The deformation is assumed to be quasi-static with kinematic nonlinearity. The material is assumed to be isotropic, elastic with Young's modulus $E = 1$ and Poisson's ratio $\nu = 0$. The mechanical deformation is driven by thermal eigenstrain:

$$\boldsymbol{\varepsilon}^{thermal} = \alpha(T - T_0)\mathbf{I}, \tag{6.1}$$

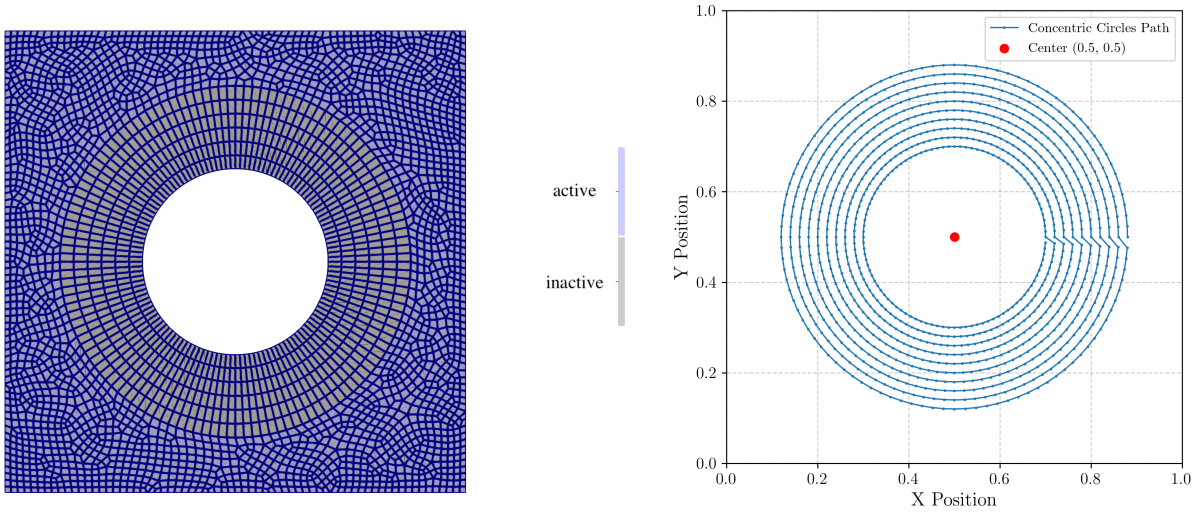
where the thermal expansion coefficient is $\alpha = 10^{-7}$ and the reference temperature is $T_0 = 0$.

All external boundaries are fixed, with $u_x = u_y = 0$ and $T = 0$. The simulations are performed with time step size of 1. Contour plots of temperature and von Mises stress are shown in Figure 6.2 and Figure 6.3, respectively.

To demonstrate the capability of modeling path-dependent material behavior, we retain the same discretization and employ the viscoplasticity material model with isotripic hardening. The material parameters are modified as follows: thermal conductivity $k = 45$, specific heat $c_p = 0.5$, and density $\rho = 8000$. The ellipsoidal heat source $q(\mathbf{x}, t)$ has semi-axes $a = 0.02\,L$ (in-plane) and $b = 0.005\,L$ (through-thickness). The total power of the heat source is set to $P = 100$, with an efficiency factor of $\eta = 1$. The thermal expansion coefficient is $\alpha = 5 \times 10^{-7}$, and the reference temperature is $T_0 = 0$. For the viscoplasticity model with linear isotropic hardening implemented in `NEML2`, we use the same parameters as described in Section 5.4. The reader is referred to Table 5.3 for the detailed parameter values used in the model.

Simulations are conducted with a time step size of 1. The resulting contours of temperature, von Mises stress, and plastic strain are shown in Figure 6.4, Figure 6.5, and Figure 6.6, respectively.

Under the applied uniaxial tension in the x-direction, the stress distribution reveals classical stress concentration effects around the circular hole. As shown in Figure 6.5, von Mises stress is predominantly concentrated along the vertical axis (i.e., the top and bottom of the hole), orthog-

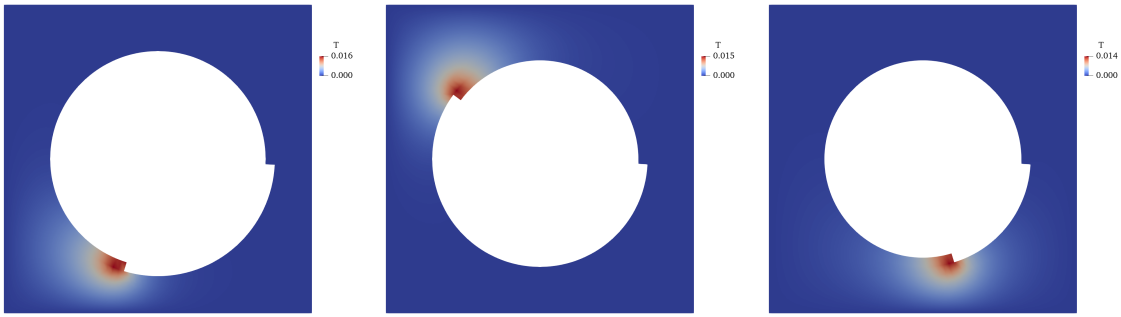


(a) (b)

Figure 6.1: Setup for the thermo-mechanical simulation of slot welding (Section 6.1). (a) Boundary-fitted mesh with linear quadrilateral elements. The *inactive* (■) and *active* (■) subdomains are shown with different colors. (b) The weld path with moving heat source.

onal to the loading direction. This pattern leads to early yielding in these regions, as confirmed by the plastic strain contours in Figure 6.6.

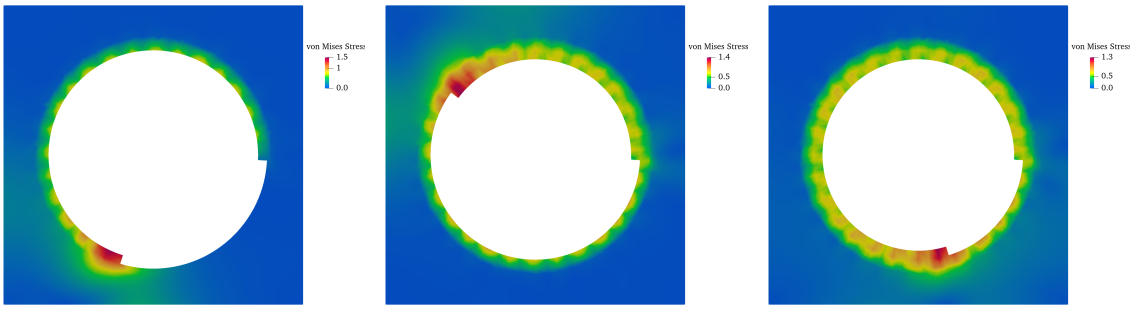
In Figure 6.7, the effect of variable initialization strategy is clearly illustrated. Figure 6.7a shows the ratio of the initial residuals, calculated as the ratio of initial residual with variable initialization using initial conditions to that with variable initialization using patch polynomial extrapolation. The ratio remains above unity for a substantial portion of the simulation, confirming the trend observed in Figure 6.7b: Patch polynomial extrapolation improves the quality of the initial guess supplied to the nonlinear solver, thereby reducing the number of iterations required for convergence. Overall, employing variable extrapolation results in approximately 12% fewer nonlinear iterations compared to the naive initialization strategy.



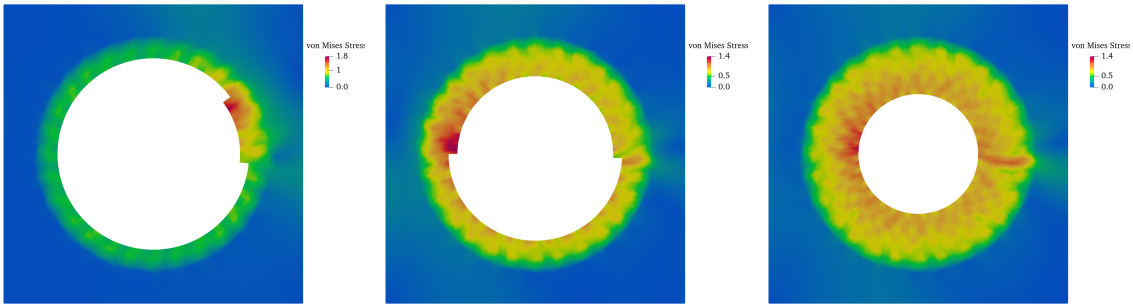
(a) $t = 70$ (b) $t = 140$ (c) $t = 180$

(d) $t = 310$ (e) $t = 450$ (f) $t = 950$

Figure 6.2: Temperature contours of the slot welding simulation with an isotropic elastic material (Section 6.1).

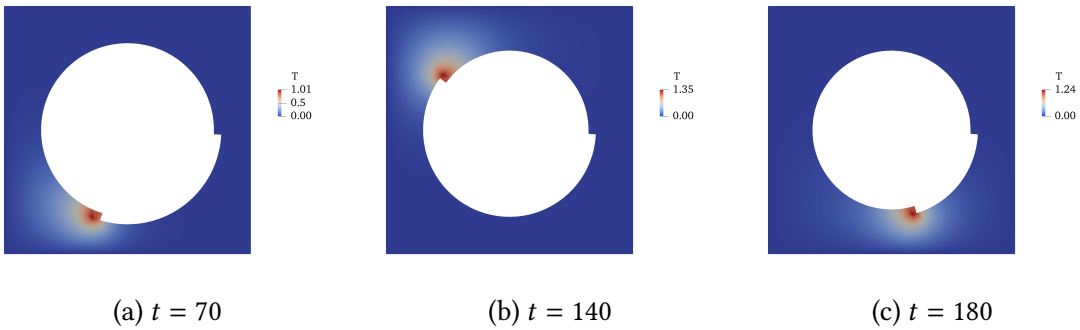


(a) $t = 70$ (b) $t = 140$ (c) $t = 180$

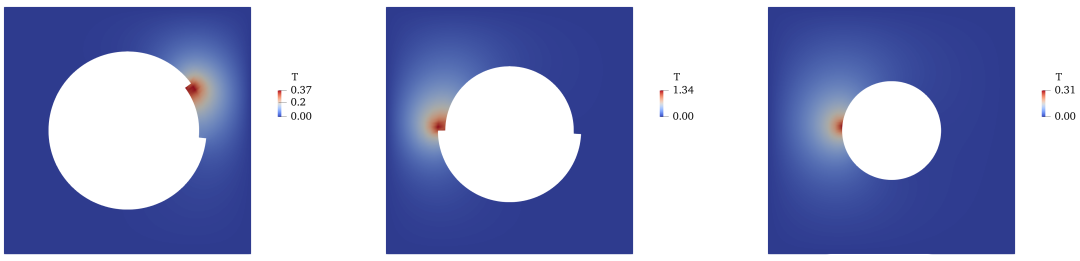


(d) $t = 310$ (e) $t = 450$ (f) $t = 950$

Figure 6.3: Contour plots showing von Mises stress during the slot welding simulation with an isotropic elastic material (Section 6.1).

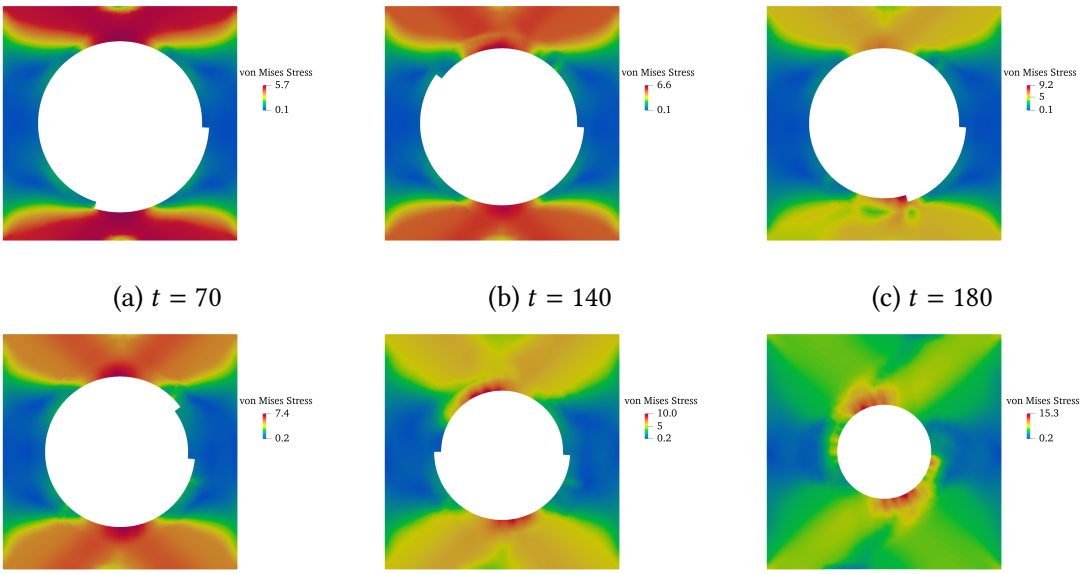


(a) $t = 70$ (b) $t = 140$ (c) $t = 180$



(d) $t = 310$ (e) $t = 450$ (f) $t = 950$

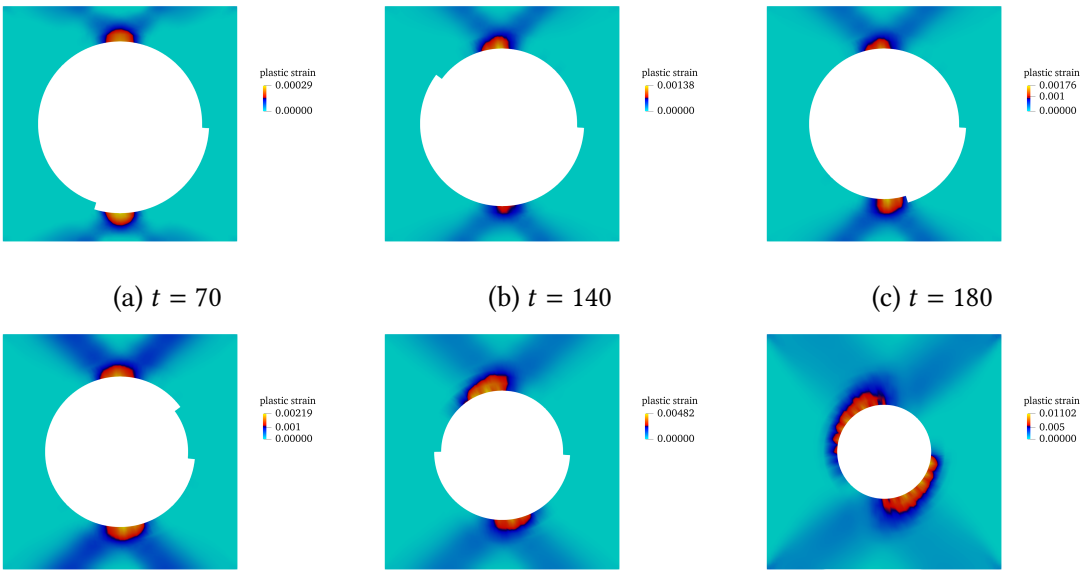
Figure 6.4: Temperature contours of the slot welding simulation with a viscoplastic material model (Section 6.1).



(a) $t = 70$ (b) $t = 140$ (c) $t = 180$

(d) $t = 310$ (e) $t = 450$ (f) $t = 950$

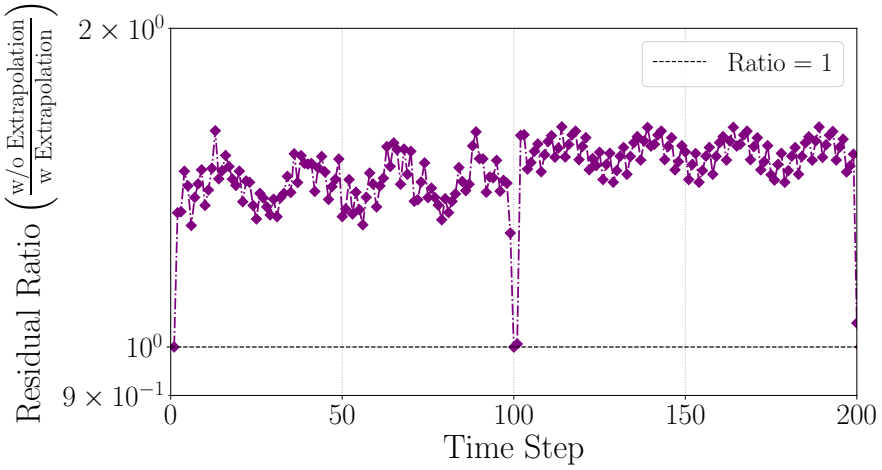
Figure 6.5: Contour plots showing von Mises stress during the slot welding simulation with a viscoplastic material model (Section 6.1).



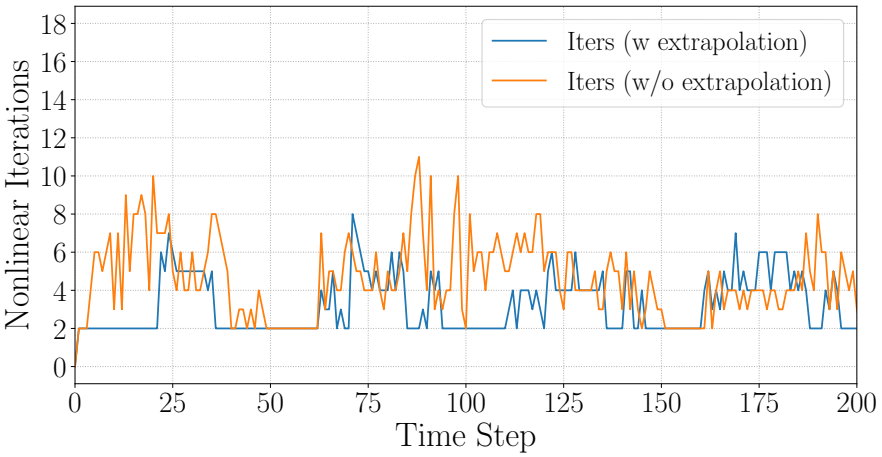
(a) $t = 70$ (b) $t = 140$ (c) $t = 180$

(d) $t = 310$ (e) $t = 450$ (f) $t = 950$

Figure 6.6: Contour plots showing equivalent plastic strain during the slot welding simulation with a viscoplastic material model (Section 6.1).



(a)



(b)

Figure 6.7: Comparison between extrapolated and non-extrapolated schemes (Section 6.1). (a) Initial-residual ratio. (b) Nonlinear iteration counts at each time step.

6.2 Three-dimensional welding simulation

Building on the previous 2D demonstration, this example presents a three-dimensional, one-way coupled thermo-mechanical simulation of the welding process.

In this example, we use the viscoplasticity material model with linear isotropic hardening summarized in Table 6.1. The computational domain consists of both *inactive* and *active* subdomains, as illustrated in Figure 6.8. The mesh resolution is set to $n_x \times n_y \times n_z = 20 \times 20 \times 40$, covering a physical domain of $[0, 2] \times [0, 0.25] \times [0, 1]$. The simulation is performed with a timestep size of 1 with a total time of 120. Dirichlet boundary conditions of $T = 0$ and $u_x = u_y = u_z = 0$ are applied on the bottom (y^-) of the domain.

At each time step, a subset of *inactive* elements is activated to mimic weld material deposition. This activation occurs progressively along the x-direction as shown in Figure 6.9b, beginning from the initial activation in Figure 6.9a. Once a full row along x is completed, the activation pattern advances upward in the y-direction to initiate the next layer (Figure 6.9c), effectively replicating a multi-pass welding process.

A moving ellipsoidal heat source $q(\mathbf{x}, t)$ with a Gaussian profile is used. The expression of the heat source is given by:

$$q(\mathbf{x}, t) = q_0 \cdot \exp\left(-(x - x_c(t))^2 - (y - y_c(t))^2 - (z - z_m)^2\right), \tag{6.2}$$

where $q_0 = 10^5$ is the peak intensity of the heat source, z_m is defined as the midpoint in the z-direction of the simulation domain (i.e., $z_m = (z_{\min} + z_{\max})/2$), and $(x_c(t), y_c(t))$ specifies the instantaneous location of the laser focal point in the xy-plane. We introduce a characteristic time scale $\Delta t = L_x/v_x$, where L_x is the scan length in the x-direction, and $v_x = 0.1$ is the weld speed along the x-axis. Within each period Δt, one full weld pass is completed from the left boundary (x^-) to the right boundary (x^+). The weld path can be expressed as:

$$x_c(t) = x_0 + v_x \cdot \mathrm{mod}(t, \Delta t), \tag{6.3}$$

$$y_c(t) = y_0 + d_y \cdot (1 + \mathrm{floor}(t/\Delta t)). \tag{6.4}$$

Here, $y_0 = 0.1$ is the initial height at which material deposition begins, and $d_y = 0.025$ denotes the layer height in the y-direction for each deposition step. $\mathrm{mod}(t, \Delta t) = t - \Delta t \cdot \mathrm{floor}(t/\Delta t)$ denotes the elapsed time within the current scan pass, enabling continuous horizontal motion of the heat source along the x-direction; $\mathrm{floor}(t/\Delta t)$ counts the number of completed horizontal passes, incrementing the vertical position by d_y after each pass to represent layer-by-layer material addition along the y-axis.

The temperature, von Mises stress, and plastic strain contours are presented in Figure 6.10, Figure 6.11, and Figure 6.12, respectively.

As shown in Figure 6.11 and Figure 6.12, high von Mises stress and plastic strain magnitudes are observed near the four bottom corners of the domain. This is attributed to the fixed displacement boundary conditions applied at the bottom surface, which restrict movement. As new hot material is deposited, thermal expansion induces displacement in the deposition region, which propagates throughout the domain, resulting in stress concentrations in these constrained areas.

For the bulk printing simulation, we also perform the same studies on the initial residual and the number of nonlinear iterations in Section 6.1, comparing the cases with and without IC extrapolation. Unlike the results in Section 6.1, the non-linear system here exhibits a higher initial residual when IC extrapolation is enabled than when it is disabled. Nevertheless, as the simulation progresses, we observe that the initial residual for the non-IC extrapolation case starts to increase, as shown in Figure 6.13a. Regarding the non-linear iteration count illustrated in Figure 6.13b, we do not observe a clear benefit from using IC extrapolation at the beginning of the simulations; the number of non-linear iterations does not decrease significantly using the IC extrapolation scheme. However, when the simulation without IC extrapolation reaches timestep 100, the non-linear iteration count grows extremely high and fails to effectively reduce the residual, ultimately causing divergence at this timestep.

As such, from both the results observed in this section (Section 6.2) and the previous section (Section 6.1), we can clearly conclude that adopting the IC extrapolation strategy effectively reduces the initial residual and prevents the simulation from diverging.

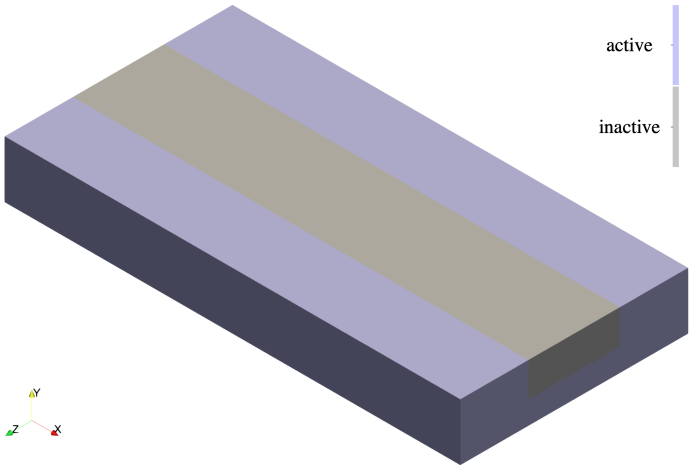
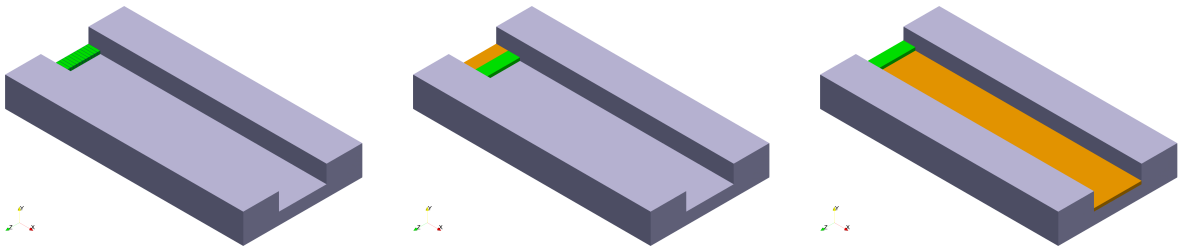


Figure 6.8: 3D computational domain and deposition demonstration.

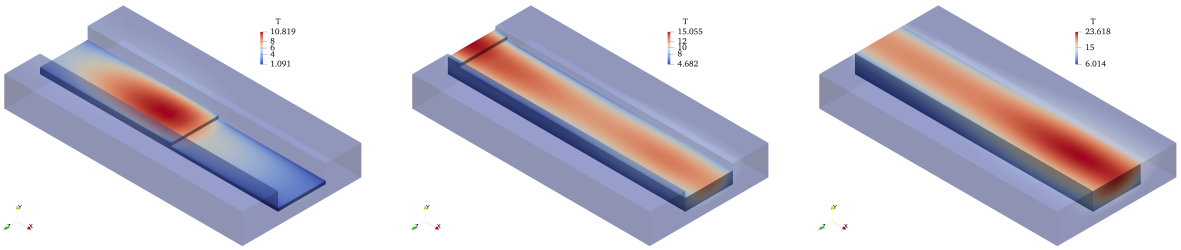


(a) Timestep $t = 1$ (b) Timestep $t = 2$ (c) Timestep $t = 21$

Figure 6.9: Decomposition of the computational domain into two types of subdomains: *active* (∪), and *updated active* ( ) elements, shown at different timesteps. The *inactive* elements are omitted (thresholded out) to better visualize the activation and deposition process.

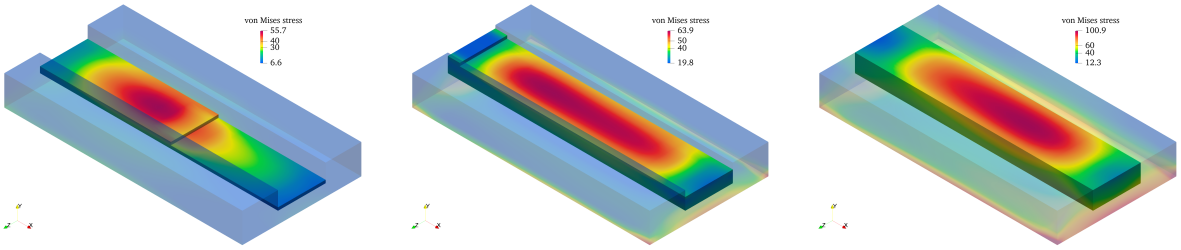
Table 6.1: Material parameters for linear isotropic hardening model

Symbol	Description	Value
E	Young's modulus	1×10^5
ν	Poisson's ratio	0.3
σ_y	Yield stress	100
H	Hardening modulus	1000
η	Perzyna reference stress	1000
n	Perzyna exponent	5



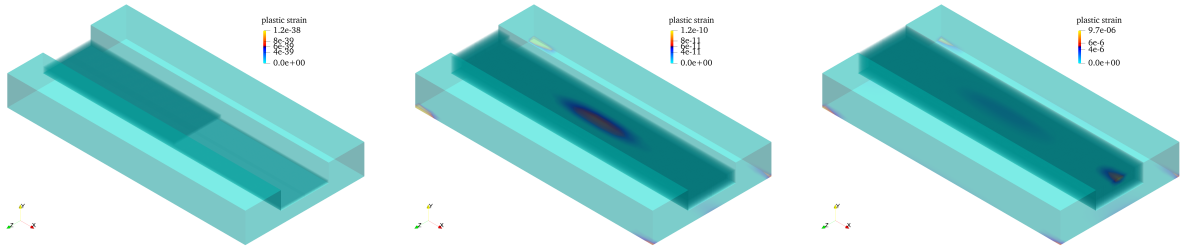
(a) Timestep $t = 30$ (b) Timestep $t = 80$ (c) Timestep $t = 120$

Figure 6.10: Temperature contour visualizations at different timesteps. *inactive* elements are omitted (thresholded out) to clearly show the activation and deposition process.



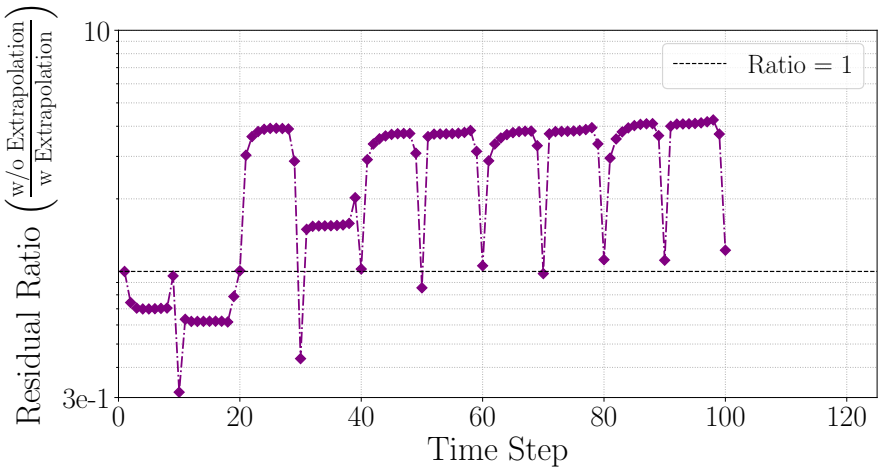
(a) Timestep $t = 30$ (b) Timestep $t = 80$ (c) Timestep $t = 120$

Figure 6.11: von Mises stress contour visualizations at different timesteps. *inactive* elements are omitted (thresholded out) to clearly show the activation and deposition process.

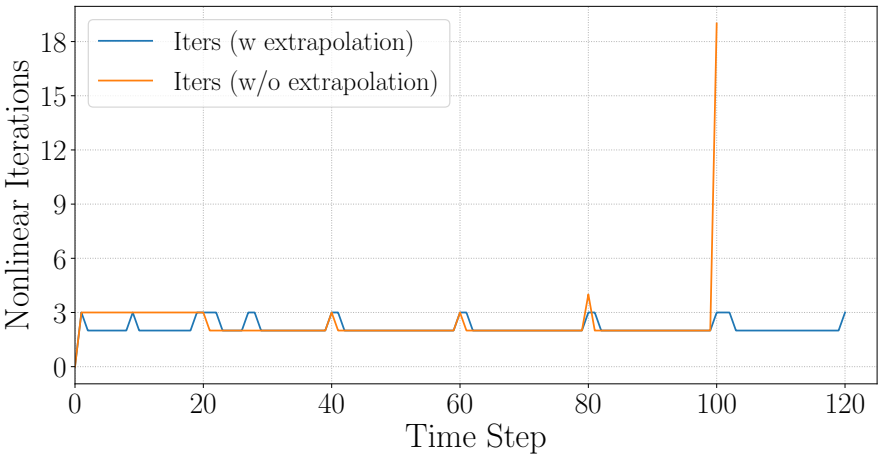


(a) Timestep $t = 30$ (b) Timestep $t = 80$ (c) Timestep $t = 120$

Figure 6.12: plastic strain contour visualizations at different timesteps. *inactive* elements are omitted (thresholded out) to clearly show the activation and deposition process.



(a)



(b)

Figure 6.13: Comparison between extrapolated and non-extrapolated schemes (Section 6.2). (a) Initial-residual ratio. (b) Nonlinear iteration counts per time step.

7 Conclusions

This report presents an enhanced simulation system for welding processes within the framework of MOOSE. By adopting a multifaceted approach, the work addresses critical challenges in user accessibility and solver robustness, particularly for simulating welding processes and related applications. The main contributions of this report are summarized as follows:

- **Enhanced user-friendliness**: A centralized default block restriction mechanism is introduced to streamline input file configuration. This eliminates the need for repetitive block specifications across different objects, thereby reducing user error and improving workflow efficiency. Additionally, a modular action for moving heat source simulations has been developed to unify path parsing, subdomain modification, and heat source kernel enforcement within a single interface.
- **Improved solver robustness**: To mitigate convergence issues caused by improper initial guesses for the DoFs associated with *updated active* elements during the simulation, the report proposes innovative initial-condition strategies. These include a weighted-averaging scheme and a patch-recovery technique that assign physically reasonable values to the new degrees of freedom by referencing data from the *stationary active* regions. This approach has proven effective in improving the convergence behavior of nonlinear material models, particularly those used in NEML2.
- **Comprehensive validation**: The proposed framework was rigorously validated through benchmark simulations, including steady-state and transient heat conduction, linear elasticity, and viscoplasticity. Results demonstrate good agreement between simulations with dynamic element activation and those using fixed meshes, confirming the framework's accuracy and reliability.
- **Large-scale application demonstrations**: Finally, the enhanced framework was applied to large-scale 2D and 3D thermo-mechanical simulations of the welding process. The resulting distributions of temperature, von Mises stress, and plastic strain align with expected physical behavior such as stress concentrations, illustrating the system's capability to handle complex problems and nonlinear material responses.

In summary, the enhancements developed and validated in this report provide a useful set of tools for the MOOSE framework in the field of advanced manufacturing. These improvements not only lower the barrier to entry for users but also increase the robustness and reliability of the simulations, laying a foundation for the prediction and optimization of residual stress and distortion in processes like welding.

Acknowledgements

The work of the Argonne authors was sponsored by the U.S. Department of Energy, under Contract No. DE-AC02-06CH11357 with Argonne National Laboratory, managed and operated by UChicago Argonne LLC.

Programmatic direction and support was provided by the Nuclear Energy Advanced Modeling & Simulation Program of the Department of Nuclear Energy. The authors gratefully acknowledge the support and direction of Ben Spencer, at Idaho National Laboratory, program director David Andersson, and Federal Manager Dave Henderson.

Bibliography

[1] T. Hu, M. C. Messner, D. Schwen, L. B. Munday, and D. Yushu, "Neml2: A high performance library for constitutive modeling," tech. rep., Argonne National Laboratory (ANL), Argonne, IL (United States); Idaho …, 2024.

[2] L. Harbour, G. Giudicelli, A. D. Lindsay, P. German, J. Hansel, C. Icenhour, M. Li, J. M. Miller, R. H. Stogner, P. Behne, *et al.*, "4.0 moose: Enabling massively parallel multiphysics simulation," *SoftwareX*, vol. 31, p. 102264, 2025.

Applied Materials Division
Argonne National Laboratory
9700 South Cass Avenue, Bldg. 212
Argonne, IL 60439

www.anl.gov