INL/RPT-22-69312 Rev 000

Implement and Test 3D Mortar Contact in BISON

September 2022

Antonio Recuero and Dewen Yushu

Idaho National Laboratory, Idaho Falls, ID 83415, USA

INL/RPT-22-69312 Rev 000

Implement and Test 3D Mortar Contact in BISON

Antonio Recuero and Dewen Yushu

Idaho National Laboratory, Idaho Falls, ID 83415, USA

September 2022

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

http://www.inl.gov

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

Page intentionally left blank

Contents

Page intentionally left blank

Milestone Title

Implement and Test 3D Mortar Contact in BISON.

Milestone Description

- **Scope**: To leverage BISON's strength as a fully 3D fuel performance code we need to take mortar contact from 2D to 3D. This would enable the modeling of asymmetric or damaged fuel pellets and their interaction with the cladding, as well as microreactor geometries. High fidelity fuel clad interaction requires the modeling of frictional contact for which we continue development of a robust mortar based implementation.
- **Objectives**: Investigate second order convergence in BISON assessment cases, research and implement fuel clad friction models, implement assessments for the verification of frictional forces, and implement 3D contact verification cases. For damaged pellet and microreactor geometries, we work towards implementing edge and vertex contact and on the verification and possible improvements of edge dropping treatment for partial contact problems. TRISO mechanical and thermal contact can benefit from these developments, and we will implement verification cases for particle fuel cases including fuel failure in partially fractured particles. Dual basis generation will be updated to support 3D mortar contact and 2D/3D edge dropping, where in both cases more frequent recomputation of the dual basis shape functions is required. We will investigate how to combine field split with the variable condensation preconditioner (VCP) to improve multiphysics performance and scalability of VCP with iterative solvers. We will review and make the current implementation consistent with the most recent literature. We aim to publish select aspects of the contact work in the open literature.

1 Background

1.1 Progress

We leverage the extension of the generation of mortar segment meshes to three dimensions in MOOSE's framework to extend thermomechanical modeling capabilities to problems with three dimensions. A modular approach to gap heat transfer physics using the mortar finite element method was created and documented, mechanical contact was extended to three dimensions—including frictional behavior, performance and ease of use were improved, and steps towards scalability of solid mechanics problems involving contact were taken. Many of these new developments are demonstrated in the simulation of 3D light-water reactor (LWR) problems, where the thermomechanical interface problem is solved using the mortar finite element method. Usage of the mortar framework has improved convergence in 2D problems and has enabled employing friction in 3D problems, of which we show results of a short, local stack of 3D pellets. Consequently, the benefits of mortar in terms of solution convergence and quality [7, 11] are extended to three dimensions.

Section 2 discusses fundamental developments that enabled the simulation of practical mortar problems in three dimensions and other general improvements, including the reduction of the derivative container size, the modification of dual basis computations when edge dropping (lack of secondary element projection) takes place, the improvement of conditioning when employing the VCP in-edge dropping conditions, and code usability and quality improvements. These latter code enhancements include the migration of tests using "old" mortar contact constraints to using dual mortar with a semi-smooth Newton solution strategy and the reuse of lower dimensional domains for straightforwardly setting up a mortar thermomechanical LWR problem, i.e. the MOOSE action is employed for mechanical contact and the thermal LWR action is employed to capture the gas conductance, contact, and radiation components of gap heat transfer physics. Independently of the mortar LWR thermal action, we developed a modular approach to gap heat transfer that resides in MOOSE and can be leveraged, e.g., in metallic fuel problems. This approach, whose code design based on MOOSE's user objects to model specific physics was proposed by the maintenance activity, is detailed in Section 3. Based on the dual mortar finite element method, the frictional contact constraints were extended to three dimensions. A block sheared in two directions in and out of contact with a rigid plane is employed in Section 4 to show the way the approach handles changes in frictional states (e.g. stick to slip) within a competitive number of Newton iterations. Equations and numerical results on the use of the VCP with Cartesian Lagrange multipliers, whose combination enables their direct condensation, are described in Section 5.3. Two-dimensional and three-dimensional BISON LWR simulations are discussed in Section 6. Particularly, a stack of five eccentric pellets with a surface defect is simulated and the effect of pellet-cladding friction is assessed. Finally, conclusions are outlined in Section 7.

1.2 Future Directions

The use of the mortar finite element method for thermal contact problems and the dual mortar approach with a semi-smooth Newton strategy for the solution of mechanical contact has demonstrated an improvement in the quality of the primal variable solution as a result of its variational consistency. We extended these benefits to three-dimensions using enforcement with Lagrange multipliers. Using Lagrange multipliers gives rise to a saddle point problem that impedes practical use of scalable iterative solvers. Strategies to leverage iterative solvers, such as preventing algebraic coarsening of the contact interface in algebraic multigrid solvers, are proposed to tackle the scalability challenge. Relaxing the enforcement of contact constraints by employing a penalty approach

removes the need for creating Lagrange multiplier variables at the expense of an enforcement with potentially lower quality. An augmented Lagrange strategy can guarantee the enforcement to a prescribed tolerance, with the caveat that it requires multiple system evaluations. Having these strategies available to the user will increase the robustness of the code and allow for tackling challenging problems involving contact, which may include nuclear reactors with arbitrary geometries in 3D.

As deformation of contacting surfaces increases, the mortar segment mesh, especially for problems in three dimensions, has shown to run into difficulties while creating segments. While this may be caused by poor initial displacement predictions, which make projections challenging, robustness of the segment generation algorithms can likely be further improved.

As demonstrated in the BISON numerical results section of this document, friction is key to determining the mechanical state of the fuel-rod system once the fuel comes into contact with the cladding. The accuracy benefits of including friction can be leveraged in fast 1.5D simulations, in which 1D layers of fuel and cladding are stacked to represent the fuel rod's height.

The items in this section are suggested as directions for future work.

2 Extension of Mortar Framework

The extension of the automatic mortar generation to the creation of three-dimensional segments allows for employing constraint and residual enforcement on segments in a mortar sense. This extension was mainly performed by the MOOSE team (Jacob Badger and Alexander Lindsay).

In this section we present improvements to the mortar finite element framework that do not fall within the scope of preconditioner, physics, or application to nuclear reactor problems.

2.1 Reduction of Automatic Differentiation Container Size

Usage of automatic differentiation (AD) allows for developing physics models for contact mechanics, gap heat transfer, and boundary conditions without having to derive tedious Jacobian entries. This is advantageous when testing new models and does not typically incur detrimental simulation slowdowns since the AD objects are solely created in lower-dimensional domains. In essence, the mortar constraints are computed on segments created from the mesh's sidesets. One practical limitation of AD is the requirement of derivative container sizes that would run over the allocated stack memory. In other words, the size of the container can become too large. Despite this situation not being typical, it happens in situations such as: (a) 3D simulations; (b) multi-physics problems (i.e. displacement, temperature, and Lagrange multiplier variables); (c) primary surface mesh significantly more refined than secondary surface; and (d) second-order Lagrange elements.

To enable general BISON simulations leveraging the mortar finite element framework, the influence of internal nodes of elements that are used to generate the lower-dimensional mesh on the Jacobian entries are eliminated. Despite these entries being negligible since they are internal to the solid, AD indices are created to hold their value, which increases the overall size of the container. A more graphical depiction is given in Fig. 1. In this task, we removed those unnecessary container entries to limit the stack memory required to run mortar AD-based mortar simulations. This procedure, applied to mortar objects, allows cutting in half the needed size for the AD container for linear elements (`HEX8`) and to a third in the case of second-order Lagrange elements (`HEX27`).

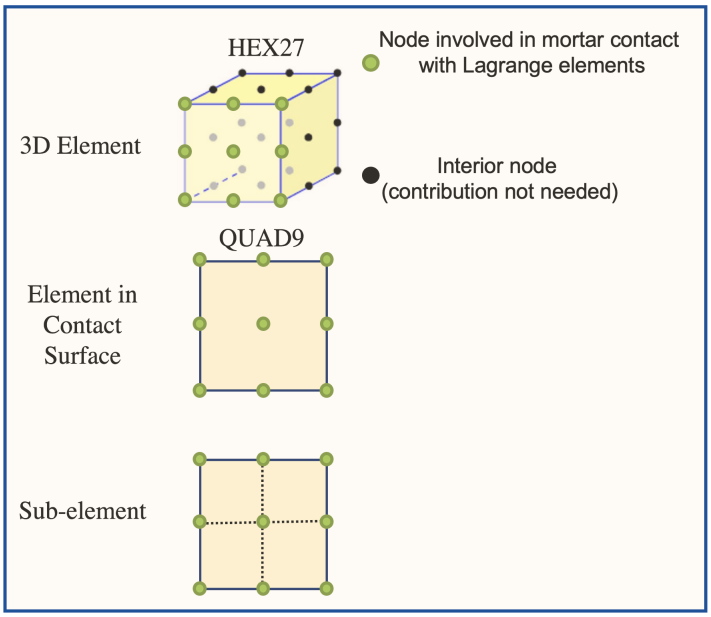


Figure 1: Removal of the constraint/residual derivatives associated with interior nodes reduces the container size required for mortar thermomechanical simulations.

The methodology described in this section is applied automatically within the code and has enabled the analysis of larger three-dimensional BISON LWR problems.

2.2 Code Cleanup

Most tests using mortar legacy contact constraints have been migrated to the current dual mortar-based contact constraints using the semi-smooth Newton method for solving the contact problem.

2.3 Mortar Thermal LWR Action

The mortar thermal LWR action and its underlying constraint were updated to accept AD materials, which allows its use in fully AD input files. In addition, the constraint was updated to account for the jump distance, thermal accommodation, and gap conductance models due to Toptan et al [13, 14].

2.4 Ease of Use

One computational feature of the mortar finite element framework in MOOSE is the need to create a lower-dimensional mesh that is used to enforce constraints and compute residuals. When using thermal and mechanical mortar contact, these lower-dimensional subdomains need to be created by one of the actions, either the mechanical or the gap heat transfer action. In BISON, the user can now use directly both actions without having to define the set of constraints and generalized forces that define contact mechanics problems. To summarize:

- In BISON, the contact mechanics and LWR thermal contact actions can be used together. The LWR thermal action leverages existing lower-dimensional domains if found in the system.

- In MOOSE, the general gap heat transfer action can be used together with the contact mechanics action by providing the primary and secondary subdomain names that contain the corresponding lower-dimensional meshes.

These developments reduce the burden on the user side. In order to migrate from a conventional input file to a mortar input file, one needs to:

- Select the mortar options for the contact mechanics and thermal contact LWR actions. This step is virtually identical to the existing node-to-segment (NTS) approach.
- Use specific PETSc options in the executioner, namely PJFNK's perturbation parameter and a non-zero shift to avoid singularity while solving the linear system.
- Block restrict heat conduction kernels. Because of the introduction of lower-dimensional domains, there is a need to restrict the blocks to which physics apply.
- Optionally, the user can leverage the reference residual solution strategy and leave out Lagrange multipliers from convergence, i.e. only converge on primal variables by using the option: `converge_on = 'displacements temperature'`.

Guidelines for usage of mortar formulations are included on the BISON website [1].

3 Extension of Thermal Contact to 3D

The extension of the automatic mortar generation to 3D problems allows for leveraging gap heat transfer in a straightforward manner. The effort of this milestone on the modeling of gap heat transfer in three dimensions has been focused on the following items:

- **Modular gap heat transfer**. A mortar constraint-based gap heat transfer design was developed in MOOSE to capture the heat flux caused by radiation, gas conduction, and mechanical contact, among other physics phenomena. The action created allows for leveraging mortar simply by providing surfaces and model parameters or by creating user-defined user objects. BISON fuel performance modeling applications, such as metallic fuel, can leverage this action to model 2D and 3D gap heat transfer. Other specialized pressure-dependent heat flux models have already been added to this modular approach in the MOOSE heat conduction module.
- **LWR gap heat transfer**. The existing mortar LWR thermal action can also leverage the lower-dimensional domains in 3D problems. A heat flux variable is enforced to have the value of the overall gap conductance, including contributions such as conduction due to mechanical contact and radiation. Small modifications to the existing gap heat transfer constraint were sufficient to extend its use to three dimensions.

Unlike for contact mechanics, same order Lagrange finite element interpolation is recommended for primal variables (temperature) and Lagrange variables.

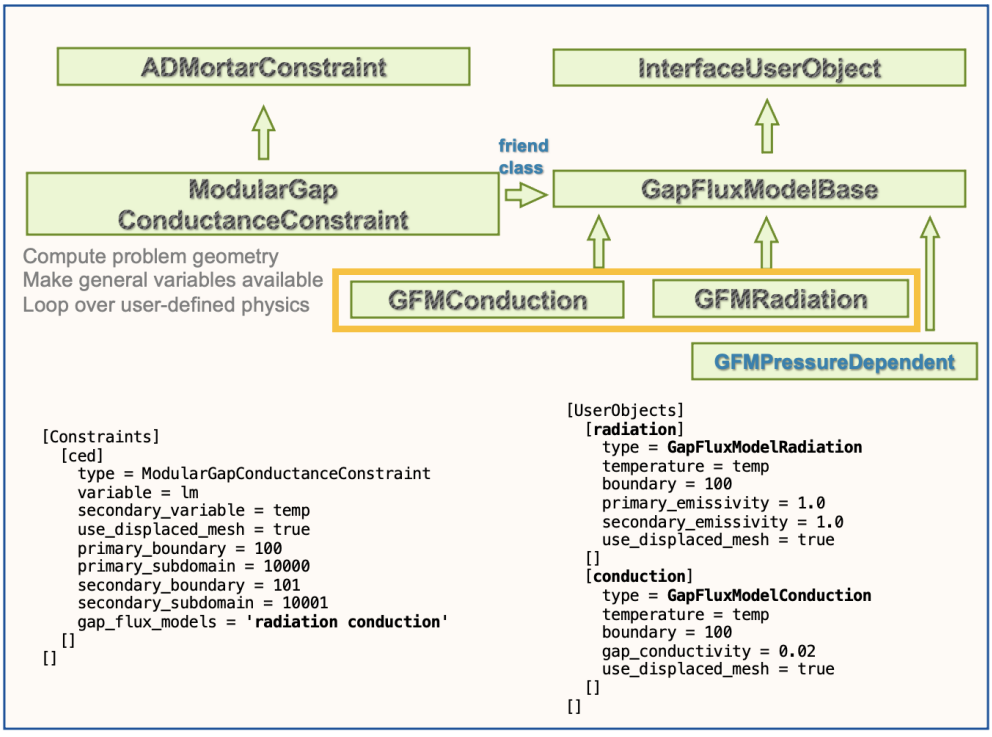


Figure 2: Diagram depicting class inheritance and input file usage of general gap heat transfer physics implemented in MOOSE. —GapFluxModelConduction— and —GapFluxModelRadiation— user objects contain physics-specific equations.

Figure 2 shows that `ModularGapConductanceConstraint` incorporates interface user objects to add physics contributions to the gap heat transfer flux (i.e. using `gap_flux_models`). Users can implement their own interface user object, which may be added to the mortar thermal contact action in MOOSE. Additionally, to capture BISON's typical use in metallic fuel, MOOSE's mortar thermal contact action allows for the automatic creation of these user objects if proper physics parameters are passed. For example, the action takes the following form when users add their own flux models implemented in user objects:

```
[MortarGapHeatTransfer]
  [mortar_heat_transfer]
   temperature = temp

   boundary = 100
   use_displaced_mesh = true

   primary_boundary = 100
   secondary_boundary = 101
   user_created_gap_flux_models = 'radiation_uo conduction_uo'
  []
[]
```

The definition of user objects (physics) is given as:

```
[UserObjects]
  [radiation_uo]
    type = GapFluxModelRadiation
    temperature = temp
    boundary = 100
    primary_emissivity = 1.0
    secondary_emissivity = 1.0
    use_displaced_mesh = true
  []
  [conduction_uo]
    type = GapFluxModelConduction
    temperature = temp
    boundary = 100
    gap_conductivity = 0.02
    use_displaced_mesh = true
  []
[]
```

This combination of the action `MortarGapHeatTransfer` and interface user objects in `UserObjects` allows the mortar constraint to include user-prescribed gap heat transfer physics. Alternatively, the user can include those two user objects directly in an action block, which will trigger the creation of those user objects upon object construction. An analogous input file would then have the following excerpt:

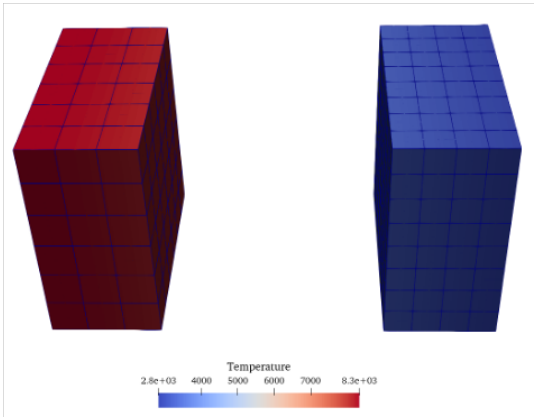
```
[MortarGapHeatTransfer]
  [mortar_heat_transfer]
   temperature = temp

   primary_emissivity = 1.0
   secondary_emissivity = 1.0
   boundary = 100
   use_displaced_mesh = true
   gap_conductivity = 0.02

   primary_boundary = 100
   secondary_boundary = 101
   gap_flux_options = 'CONDUCTION RADIATION'
  []
[]
```

Verification of the implementation against the existing NTS approach, for typical finite element such as HEX8, HEX20, and HEX27, was performed. Flat and curved geometries were used (see Fig. 3a for two regression tests available in MOOSE).

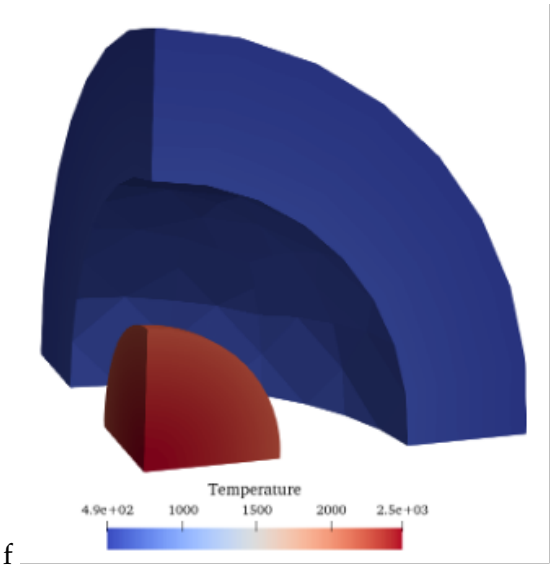
Results show good agreement between the mortar approach and the surface integration approach (see Fig. 4).

To summarize:

- A general framework to perform gap heat transfer modeling leveraging the mortar finite element method was implemented and tested in MOOSE.
- The extension of mortar segment generation to three-dimensions can be leveraged in BISON for LWR and metallic fuel applications.
- Additional models can be added as needed.



(a) Gap heat transfer between two flat blocks.



f

(b) Gap heat transfer between two concentric spherical geometries

Figure 3: Two examples of regression tests used to verify the three-dimensional mortar segment building framework in conjunction with gap heat transfer via the thermal contact action.

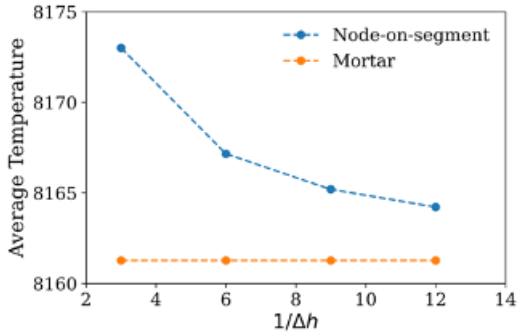


Figure 4: Convergence on the temperature variable for the flat block problem of Figure 3b with h-refinement.

4 Extension of Mechanical Contact to 3D

Here, we briefly summarize the equations necessary to perform the extension of mechanical contact to three dimensions. The dual mortar with a semi-smooth Newton approach implemented last fiscal year was extended and verified in three dimensions. We assume here discretization of mechanical contact constraints and usage of dual shape functions for Lagrange multiplier interpolation (see [18] for variational equations).

The discretized contact constraints for normal and tangential components may be expressed as [9]:

$$\begin{aligned} \left(\tilde{g}_n\right)_j \geq 0,\ \left(\lambda_n\right)_j \geq 0,\ \left(\tilde{g}_n\right)_j \left(\lambda_n\right)_j = 0, \\ \Phi_j := \left|\left|\left(\boldsymbol{\lambda}_\tau\right)_j\right|\right| - \mu \left|\left(\lambda_n\right)_j\right| \leq 0, \\ \left(\tilde{\boldsymbol{v}}_{\tau,\mathrm{rel}}\right)_j + \beta_j \left(\boldsymbol{\lambda}_\tau\right)_j = \mathbf{0},\ \beta_j \geq 0,\ \Phi_j \beta_j = 0, \end{aligned} \tag{1}$$

where $j = 1 \ldots n_\lambda$, where n_λ is the number of Lagrange multipliers, $\left(\tilde{g}_n\right)_j$ is the discrete weighted normal gap and $\left(\tilde{\boldsymbol{v}}_{\tau,\mathrm{rel}}\right)_j$ is the discrete weighted relative tangential velocity vector at the secondary node j; λ_n is the normal component of the Lagrange multipliers enforcing mechanical contact, μ is the Coulomb's friction coefficient, $\boldsymbol{\lambda_\tau}$ is the vector of Lagrange multipliers that captures contact pressure due to frictional forces, and β_j is a semi-positive variable. Specifically, weighted quantities are obtained by performing numerical integration over mortar segments and assigning corresponding values to the nodes on the secondary surface:

$$\begin{aligned} \left(\tilde{g}_n\right)_j &= \int_{\Gamma_c^{(1)}} \psi_j g_n \, \mathrm{d}s, \\ \left(\tilde{\boldsymbol{v}}_{\tau,\mathrm{rel}}\right)_j &= \int_{\Gamma_c^{(1)}} \psi_j \mathbf{v}_{\tau,\mathrm{rel}} \, \mathrm{d}s. \end{aligned} \tag{2}$$

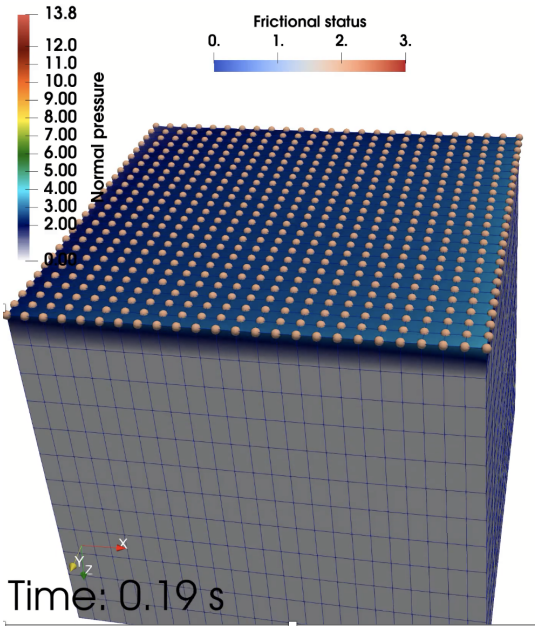
Extension to three-dimensions implies that normal and tangential vectors have a dimensionality of three. In addition, tangential forces act on two perpendicular directions which are obtained via an orthogonalization procedure [6]. The discretized contact constraints are then converted into a form that allows for a separation of contact states into branches that define states: not in contact, in contact and slipping, and in contact and sticking. As such, the discretized constraints of the normal contact constraints in Eq. 1 become:

$$\min\left(\lambda_n, c_n \left(\tilde{g}_n\right)_j\right), \tag{3}$$

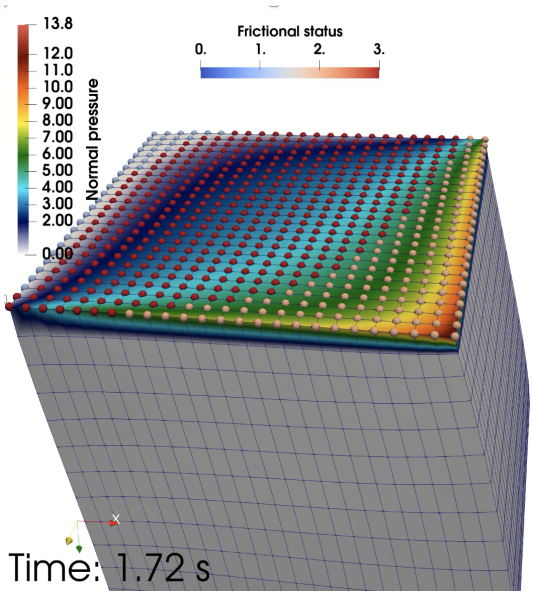
for the non-interpenetration condition, and

$$\max\left(\mu(\lambda_n + c_n \left(\tilde{g}_n\right)_j), ||\boldsymbol{\lambda_\tau} + c_t \tilde{\boldsymbol{v}}_{\tau,\mathrm{rel}}||\right)\boldsymbol{\lambda_\tau} - \mu \max\left(0, (\lambda_n + c_n \left(\tilde{g}_n\right)_j)(\boldsymbol{\lambda_\tau} + c_t \tilde{\boldsymbol{v}}_{\tau,\mathrm{rel}}\right) \tag{4}$$

for Coulomb constraints [4]. Both constraint equations are solved as a part of MOOSE's constraint system iteratively. Parameters c_n and c_t are used to weigh the gap and relative tangential displacement, respectively. The weighing of these variables is key to solving the constraint equations, as they bring dissimilarly scaled physics to comparable quantities. The role of these parameters is, therefore, to enable good conditioning of the constraint, but they do not affect the converged solution. Typically, the value of those parameters can range several orders of magnitude without severely affecting convergence behavior.



(a) Contact states of nodes are sticking ($t = 0.19\,\text{s}$).



(b) Contact states of nodes encompass not in contact, sticking, and slipping ($t = 1.72\,\text{s}$).

Figure 5: Deformable block coming into contact with a stiff "infinite" body. Initially, nodes come in contact with a stiff surface and stick (orange spheres) due to weak loading in the tangential directions. As the simulation progresses, shear-inducing boundary conditions cause several sets of contact states, including not in contact (very light blue) and slipping (red). The auxiliary kernel `MortarFrictionalState` is used to identify the contact states.

The extension of frictional mechanical contact constraints to three-dimensions was benchmarked in a problem featuring a flexible block that is driven into mechanical contact and then released while sheared in the two frictional directions. This simple example allows for exercising the three possible local contact states: not in contact, contact and sticking, and contact and slipping. The block has the following dimensions: $0.5\,\text{m} \times 0.5\,\text{m} \times 0.5\,\text{m}$; the friction coefficient is chosen to be 0.4, the material is considered to be elastic, and is assumed to make contact with a much stiffer body. The boundary conditions are such that the block is vertically compressed a maximum of $0.04\,\text{m}$ sinusoidally, and, along the X and Y directions, the $0.01t$ and $0.06t$, where t is time (a similar setup was used by Hueber [4]). That is, given these conditions, it is expected the block will go from a state of globally sticking to a state of globally slipping, making local transitions as the simulation progresses. Two snapshots of this simulation are given in Fig. 5, where it can be observed that nodes with largest normal contact force tend to stick, whereas the local material points undergoing relatively lighter normal forces tend to slip.

This example exhibits the entire array of contact states, which makes it appropriate to study the convergence behavior. For two simulations, we monitor the number of Newton iterations per fixed time step and the number of nodes in contact. For one run, the block has a modulus of elasticity of 10^2 Pa and a Poisson ratio of 0.0 (referred to as "soft"); whereas the other run features a block with a modulus of elasticity of 10^6 Pa and a Poisson ration of 0.3 (referred to as "stiff"). See Fig. 6 for details.

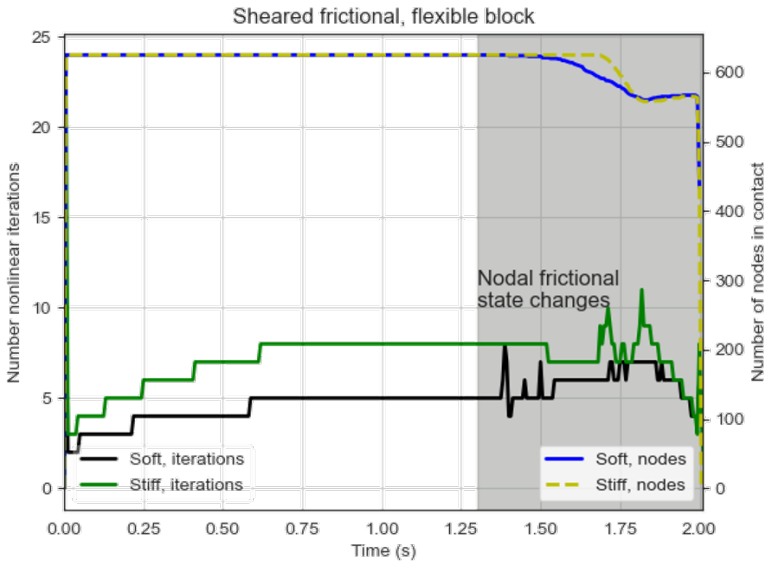


Figure 6: Number of global system's Newton iterations needed to solve the contact mechanics problem from Figure 5 and number of nodes into contact for two runs. The gray area on the plot refers to the time span when nodes switch states from in contact and stick to in contact and slip and not in contact. The number of nonlinear iterations becomes more changeable, and slightly higher, as the system requires switches in the contact states of the nodes.

Convergence results show that the dual mortar approach with a semi-smooth Newton solution strategy arrives at nodal frictional solutions within a competitive number of nonlinear iterations for a general solid mechanics problem with contact.

5 Dual Mortar and Preconditioner

This section discusses improvements that were made for the dual mortar approach, conditioning advantages that were brought by the variable condensation preconditioner (VCP), and changes in the mortar-based mechanical contact framework that were aimed at further improving the performance of the VCP.

5.1 Dual Mortar with Edge Dropping

During last fiscal year, the dual mortar approach was implemented for thermal and mechanical contact problems using first and second order Lagrange elements [18]. The dual mortar approach uses a dual shape/basis function that satisfies the following bi-orthogonal condition:

$$\int_{\Gamma_c^{(1)}} \psi^{(j)}(\boldsymbol{x})\phi^{(k)}(\boldsymbol{x})\,\mathrm{d}s = \delta_{jk}\int_{\Gamma_c^{(1)}} \phi^{(k)}(\boldsymbol{x})\,\mathrm{d}s, \tag{5}$$

where $\psi^{(j)}(\boldsymbol{x})$ is the dual shape function, $\phi^{(k)}(\boldsymbol{x})$ is the standard shape function, and δ_{jk} is the Kronecker delta function. The above condition is evaluated element-wise on each secondary surface element e, i.e.,

$$\int_{e} \psi^{(j)}(\boldsymbol{x})\phi^{(k)}(\boldsymbol{x})\,\mathrm{d}s = \delta_{jk}\int_{e} \phi^{(k)}(\boldsymbol{x})\,\mathrm{d}s. \tag{6}$$

Equation (6) requires that all secondary elements are fully contained in the contact area. For contact simulations, however, elements may not always fully participate in the contact integration area. In fact, this lack of contribution is common when one material body slides off another. This is called the edge-dropping problem. It is found that a modified calculation of the dual basis function is necessary for the correct treatment of the edge-dropping problem. Examples of the incorrect and correct edge-dropping case treatments are included in Figure 7. Here, Figure 7a calculates the dual shape functions using Equation (6) for all secondary elements, while Figure 7b uses modified dual shape functions for the secondary elements that are not fully in contact.

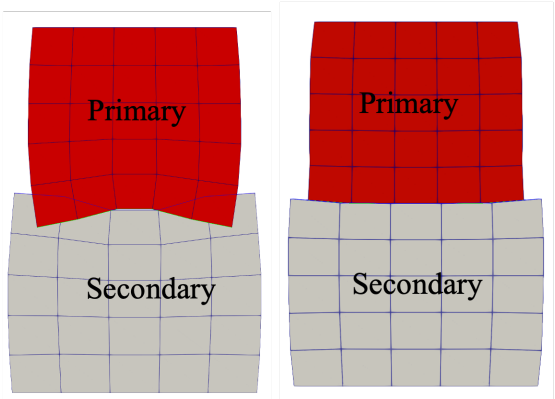


Figure 7: Examples of (a) incorrect and (b) correct edge-dropping case treatments.

We utilize a schematic of a typical edge-dropping case to illustrate the differences between the standard and modified dual shape functions (see Figure 8). Here, e_1 is fully contained in the contact area, while only a part of e_2 is contained in the contact area (i.e., e_{int}). In this case, to ensure the diagonality of the mortar matrix, the integration limits in Equation (6) needs to be replaced

by the fraction e_{int} of the secondary element that actually exists in the contact area. Therefore, the new bi-orthogonal condition for a partially in-contact secondary element becomes

$$\int_{e_{\text{int}}} \tilde{\psi}^{(j)}(\boldsymbol{x})\phi^{(k)}(\boldsymbol{x})\,\mathrm{d}s = \delta_{jk}\int_{e_{\text{int}}} \phi^{(k)}(\boldsymbol{x})\,\mathrm{d}s, \tag{7}$$

where the $\tilde{\psi}^{(j)}(\boldsymbol{x})$ represents the modified dual shape functions.

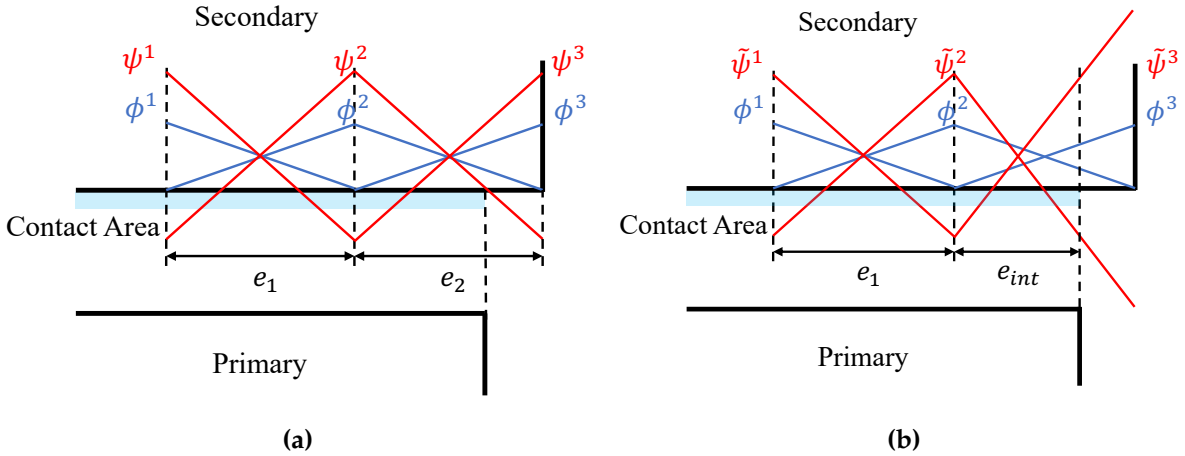
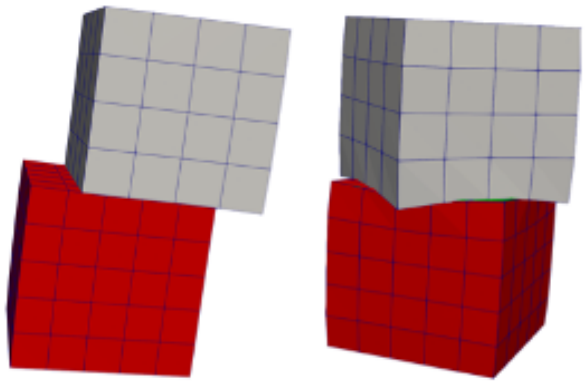


Figure 8: Schematic of a 2D edge-dropping case and dual basis functions. (a) Standard element-based dual shape functions. (b) Modified dual shape functions [10].

Although we only illustrate the modified dual basis using a 2D case (Figure 8), the definition in Equation (7) is valid for both 2D and 3D cases. In a 2D setup, the contact surface consists of 1D mortar elements e with a local coordinate $\xi \in [-1, 1]$. The beginning and the end of the contact integration area e_{int} can thus be identified by two parameters ξ_{min} and ξ_{max}, and the integration in Equation (7) can simply be performed via Gauss quadrature on the interval $[\xi_{\text{min}}, \xi_{\text{max}}]$ (Ref. [10]). In 3D, determination of the element contact integration area e_{int} becomes more complex. To evaluate the integrals in Equation (7), the primary surface is first projected onto the secondary surface and the overlap between each primary-secondary element pair is triangulated to obtain integration cells. Fortunately, the projection, overlap detection and triangulation procedure is the same typically used for the creation of mortar segments and evaluation of the mortar matrices in 3D. Therefore, the existing routine can be reused for the modified dual basis calculation. The corresponding changes to the modified dual basis calculation have been made correspondingly in libMesh [5] and MOOSE [8] and tested for first and second order elements with various quadrature rules. Figure 9 shows 3D examples of the correct edge-dropping treatments which was not possible without the implementation of the modified dual basis functions.



(a) (b)

Figure 9: Examples of correct edge-dropping cases in 3D.

5.2 System Conditioning Improvement

The edge dropping condition tends to result in ill-conditioned systems of equations. This is because when the contribution of a secondary element to the contact area decreases, the corresponding values in the mortar matrices decrease as well. These small values are then assembled into the global system of equations and may yield an ill-conditioned system.

To resolve this issue, researchers have proposed various scaling methods, i.e., element scaling [3] and nodal scaling [10]. Despite differences among different scaling strategies, the general idea of the scaling methods is similar. That is, to scale the mortar matrix based on a scale factor specially computed for each individual secondary element/node so that the entries of the scaled mortar matrices remain of constant order of magnitude, despite the change in actual contact element integration areas e_{int}.

In this work, instead of implementing an additional scaling strategy that is specifically for contact, we propose the usage of the existing VCP that was developed last fiscal year based on the MOOSE framework (see [18]). The VCP focuses on creating and solving a condensed system of equations that only has the displacement DoFs (i.e. no Lagrange multipliers in the system), which is enabled by calculating the Lagrange multipliers independently using a direct solver. Here, the separate treatment of the Lagrange multipliers from the rest of the system mitigates the ill-conditioning effects from the mortar matrices.

As a demonstration, we include an example of two identical cubes (edge length 1.0) being compressed by a vertical displacement 0.05 of the upper surface within 8 timesteps. Then the upper cube gradually slides over the right edge of the lower cube. The prescribed displacement 0.3 is achieved in another 32 timesteps. At the end, one row of secondary elements completely drops off the right edge. This example mimics what is described in [10] for a direct comparison. Snapshots of the two blocks at the beginning and at the end of the simulation are included in Figures 10a and 10b. In Figure 10c, we show the condition number change during the simulation for both the original system (blue curve) and using VCP (orange curve). It can be observed that for the original system (blue curve) the condition number increases by several orders of magnitudes (from 10^6 to 10^{11}) as the secondary element is about to slide off of the right edge. On the contrary, VCP (orange curve) greatly reduces the magnitude of this jump (from 10^4 to 10^5). Meanwhile, it is also apparent that the VCP tends to improve the overall conditioning of the system by at least two orders of magnitude throughout the entire simulation. The results shown here in Figure 10c are very close to what is presented in [10] for the scaling approaches.

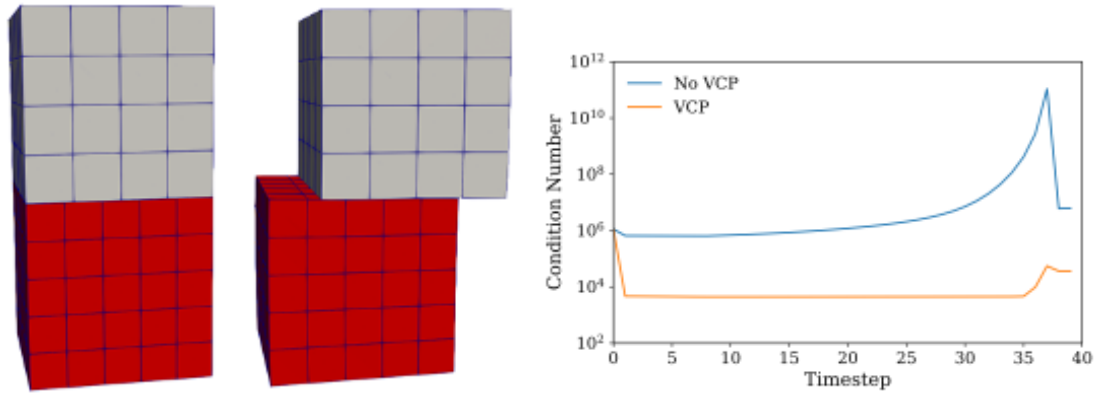


Figure 10: Two block sliding example in 3D. Deformed states at (a) timestep 0 and (b) timestep 40. (c) Condition number as a function of the timestep number.

5.3 VCP with Cartesian Lagrange Multipliers

In addition to using VCP on existing and new problems for improved convergence (see [18] and Section 5.2), we further improved the VCP this fiscal year to enable oblique contact interfaces, which is common for contact bodies that have complex geometries. This is achieved by switching the Lagrange multiplier variables' definition from the local coordinate system to the Cartesian coordinate system.

To illustrate the reason behind this change, let us start with the discretized system of equations for a typical mortar-based mechanical contact problem. The block-wise Jacobian matrix of the problem is as follows (ref. [18]),

$$\begin{bmatrix} \boldsymbol{K}_{\omega\omega} & \boldsymbol{K}_{\omega m} & \boldsymbol{K}_{\omega i} & \boldsymbol{K}_{\omega a} & 0 & 0 \\ \boldsymbol{K}_{m\omega} & \boldsymbol{K}_{mm} & \boldsymbol{K}_{mi} & \boldsymbol{K}_{ma} & -\boldsymbol{M}_i^{\intercal} & -\boldsymbol{M}_a^{\intercal} \\ \boldsymbol{K}_{i\omega} & \boldsymbol{K}_{im} & \boldsymbol{K}_{ii} & \boldsymbol{K}_{ia} & \boldsymbol{D}_i & 0 \\ \boldsymbol{K}_{a\omega} & \boldsymbol{K}_{am} & \boldsymbol{K}_{ai} & \boldsymbol{K}_{aa} & 0 & \boldsymbol{D}_a \\ 0 & 0 & 0 & 0 & \boldsymbol{I}_i & 0 \\ 0 & \widetilde{\boldsymbol{M}}_a & \boldsymbol{S}_{ai} & \boldsymbol{S}_{aa} & 0 & 0 \\ 0 & 0 & \boldsymbol{F}_{ai} & \boldsymbol{F}_{aa} & 0 & \boldsymbol{T}_a \end{bmatrix} \begin{bmatrix} \Delta\boldsymbol{u}_\omega \\ \Delta\boldsymbol{u}_m \\ \Delta\boldsymbol{u}_i \\ \Delta\boldsymbol{u}_a \\ \Delta\boldsymbol{\lambda}_i \\ \Delta\boldsymbol{\lambda}_a \end{bmatrix} = \begin{bmatrix} \Delta\boldsymbol{r}_\omega \\ \Delta\boldsymbol{r}_m \\ \Delta\boldsymbol{r}_i \\ \Delta\boldsymbol{r}_a \\ 0 \\ \Delta\boldsymbol{g}_a \\ 0 \end{bmatrix}, \tag{8}$$

where the right-hand side denotes the change in the residual, due to contact. $\boldsymbol{g}$ contains the residual entries associated with a weighted gap. $\boldsymbol{u}$ denotes the vector of displacement degrees of freedom (DoFs), and $\boldsymbol{\lambda}$ denotes the vector of the Lagrange multipliers. The subscripts m, i, a, and ω indicate the primary (mortar) side, the inactive secondary (non-mortar) side, the active secondary (non-mortar) side, and the rest of the domain, respectively.

The VCP carries out a static condensation step that eliminates the discrete Lagrange multipliers at negligible computational cost. This is achieved by a direct solve of the Lagrange multipliers as follows using the highlighted entries from Equation (8):

$$\begin{bmatrix} \Delta\boldsymbol{\lambda}_i \\ \Delta\boldsymbol{\lambda}_a \end{bmatrix} = \begin{bmatrix} \boldsymbol{D}_i & 0 \\ 0 & \boldsymbol{D}_a \end{bmatrix}^{-1} \left(\begin{bmatrix} \Delta\boldsymbol{r}_i \\ \Delta\boldsymbol{r}_a \end{bmatrix} - \begin{bmatrix} \boldsymbol{K}_{i\omega} \\ \boldsymbol{K}_{a\omega} \end{bmatrix} \Delta\boldsymbol{u}_\omega - \begin{bmatrix} \boldsymbol{K}_{im} \\ \boldsymbol{K}_{am} \end{bmatrix} \Delta\boldsymbol{u}_m - \begin{bmatrix} \boldsymbol{K}_{ii} & \boldsymbol{K}_{ia} \\ \boldsymbol{K}_{ai} & \boldsymbol{K}_{aa} \end{bmatrix} \begin{bmatrix} \Delta\boldsymbol{u}_i \\ \Delta\boldsymbol{u}_a \end{bmatrix} \right). \tag{9}$$

Note that, for Equation (9) to be valid, the highlighted block-diagonal matrix needs to be invertible, which indicates that the number of DoFs for $[\Delta\boldsymbol{\lambda}_i, \Delta\boldsymbol{\lambda}_a]^{\intercal}$ and $[\Delta\boldsymbol{u}_i, \Delta\boldsymbol{u}_a]^{\intercal}$ needs to be identical. In

other words, the number of DoFs for the displacement and the Lagrange multiplier variables should be equal on the secondary side of the contact interface. However, this is not the case in our mortar-based contact framework. The nodal number of DoFs for the displacement variable is dependent on the dimension of the problem, i.e., 2 DoFs per node for 2D problems, 3 DoFs per node for 3D problems. On the other hand, the nodal number of DoFs for the Lagrange multiplier variable is dependent on both the physics and the problem dimension, i.e., 1 DoF per node for frictionless problems, 2 DoFs per node for 2D frictional problems, and 3 DoFs per node for 3D frictional problems. To resolve this inconsistency, changes are made in the definition of the Lagrange multiplier variables and the contact constraints in order to make Equation (9) valid for all possible cases.

5.3.1 Cartesian Lagrange Multipliers

In our previous mortar formulation, the discrete nodal values of the contact traction vector, denoted by $\boldsymbol{z}_j \in \mathbb{R}^3$, is defined differently for different contact problems, i.e.,

$$\begin{aligned} &\text{Frictionless contact:} && \boldsymbol{z}_j = (\lambda_n)_j \boldsymbol{n}_j \\ &\text{Frictional contact in 2D:} && \boldsymbol{z}_j = (\lambda_n)_j \boldsymbol{n}_j + (\lambda_\tau^1)_j \boldsymbol{\tau}_j^1 \\ &\text{Frictional contact in 3D:} && \boldsymbol{z}_j = (\lambda_n)_j \boldsymbol{n}_j + (\lambda_\tau^1)_j \boldsymbol{\tau}_j^1 + (\lambda_\tau^2)_j \boldsymbol{\tau}_j^2, \end{aligned} \tag{10}$$

where λ_n, λ_τ^1, and λ_τ^2 are among the unknowns to be solved for each type of contact problem (see [18]). Specifically, λ_n is the Lagrange multiplier capturing the contact traction component that is normal to the contact surface, $\boldsymbol{n}$ is the local normal vector. For frictional problems, the λ_τ^1 is the tangential component that is aligned with the contact surface, $\boldsymbol{\tau}^1$ is the corresponding tangential vector. In three-dimensional frictional problems, an additional component, λ_τ^2 exist along the contact surface, which is associated with the second tangential vector $\boldsymbol{\tau}^2$.

Condensing Lagrange multipliers in the system of equations represents a first step towards the use of lower memory-consuming iterative solvers [9, 15]. The formulation in Equation (10), however, causes serious ambiguity while using the VCP (see [18]) during the static condensation, where one Lagrange multiplier component can correspond to multiple displacement DoFs. Besides, condensing Lagrange multipliers requires having contact tractions and associated Jacobian entries available at the condensation stage. While normal and tangential contact tractions can be computed from the existing local definition of Lagrange multipliers (see Fig. 11) by projecting the local Lagrange multipliers along local geometry vectors defined in the global frame of reference, this methodology heavily complicates the implementation of the existing VCP, as it would require it to retrieve information from mortar objects in the MOOSE framework.

Alternatively, a cleaner mortar contact mechanics implementation that works with the VCP has been carried out. This new implementation depends on the decomposition of the contact traction vector $\boldsymbol{z}_j$ in the Cartesian coordinate system, i.e.,

$$\boldsymbol{z}_j = (\lambda_x)_j (\boldsymbol{e}_x)_j + (\lambda_y)_j (\boldsymbol{e}_y)_j + (\lambda_z)_j (\boldsymbol{e}_z)_j, \tag{11}$$

where λ_x, λ_y, λ_z are the Lagrange multiplier variables along the x, y, z directions, respectively. Here, the λ_z vanishes for two-dimensional problems. The $\boldsymbol{e}_x$ $\boldsymbol{e}_y$, $\boldsymbol{e}_z$ are the unit vectors along the three directions. Compared to Equation (10), this formulation treats λ_x, λ_y, λ_z as unknowns instead. Therefore, the nodal DoF number of Lagrange multiplier variables becomes consistent with that of the displacement variables, which makes Equation (9) valid for all contact problems. As a result, this new formulation enables straightforward static condensation during the preconditioning and solution stages in the VCP.

Because of the formulation change from Equation (10) to Equation (11), an associated implementation of frictionless and frictional mechanical contact constraints was carried out, using projections of the contact traction vector from the Catesian to the local coordinate system,

$$(p_n)_j = \boldsymbol{z}_j \cdot \boldsymbol{n}_j,\ (p_\tau^1)_j = \boldsymbol{z}_j \cdot \boldsymbol{\tau}_j^1,\ (p_\tau^2)_j = \boldsymbol{z}_j \cdot \boldsymbol{\tau}_j^2, \tag{12}$$

where $(p_n)_j$, (p_τ^1), and $(p_\tau^2)_j$ are contact tractions projected in the local contact frame. Note that each of these locally projected tractions depend, in general, on three Cartesian Lagrange multipliers for a 3D problem. Equation (12) also shows the way we retrieve the normal and tangential contact traction components. An auxiliary kernel named `MortarPressureComponentAux` has been implemented to obtain local contact traction values from Cartesian Lagrange multipliers.

By substituting Equation (11) into Equation (12), we can see that p_n, p_τ^1, and p_τ^2 are now functions of λ_x, λ_y, λ_z [1]

$$\begin{aligned}
(p_n)_j &= (\lambda_x)_j(\boldsymbol{e}_x)_j \cdot \boldsymbol{n}_j + (\lambda_y)_j(\boldsymbol{e}_y)_j \cdot \boldsymbol{n}_j + (\lambda_z)_j(\boldsymbol{e}_z)_j \cdot \boldsymbol{n}_j, \\
(p_\tau^1)_j &= (\lambda_x)_j(\boldsymbol{e}_x)_j \cdot \boldsymbol{\tau}_j^1 + (\lambda_y)_j(\boldsymbol{e}_y)_j \cdot \boldsymbol{\tau}_j^1 + (\lambda_z)_j(\boldsymbol{e}_z)_j \cdot \boldsymbol{\tau}_j^1, \\
(p_\tau^2)_j &= (\lambda_x)_j(\boldsymbol{e}_x)_j \cdot \boldsymbol{\tau}_j^2 + (\lambda_y)_j(\boldsymbol{e}_y)_j \cdot \boldsymbol{\tau}_j^2 + (\lambda_z)_j(\boldsymbol{e}_z)_j \cdot \boldsymbol{\tau}_j^2.
\end{aligned} \tag{13}$$

Therefore, when imposing constraints (see Equation (1)) with the Cartesian Lagrange multipliers, each normal or tangential contact constraint will depend on all three nodal Lagrange multipliers (for a three-dimensional problem). Note, for frictionless simulations with Cartesian Lagrange multipliers, a pair of nodal constraints complement the normal contact constraint to ensure that the tangential contact traction vector is identically zero, i.e. $(p_\tau^1)_j = (p_\tau^2)_j = 0$. Upon convergence, the local Lagrange multiplier approach retrieves the contact traction expressed in the local frame, i.e.: $(\lambda_n)_j = (p_n)_j$, $(\lambda_\tau^1)_j = (p_\tau^1)_j$, and $(\lambda_\tau^2)_j = (p_\tau^2)_j$. The approach introduced in this section to enforce mechanical contact constraints therefore increases the number of necessary contact constraints in frictionless problems.

These new constraints are available in MOOSE within the objects `ComputeWeightedGapCartesianLM-MechanicalContact` and `ComputeFrictionalForceCartesianLMMechanicalContact` for the frictionless and frictional options, respectively.

[1] And, upon convergence, p_n, p_τ^1, and p_τ^2 will be identical to the Lagrange multipliers defined in the local frame, i.e : λ_n, λ_τ^1, and λ_τ^2

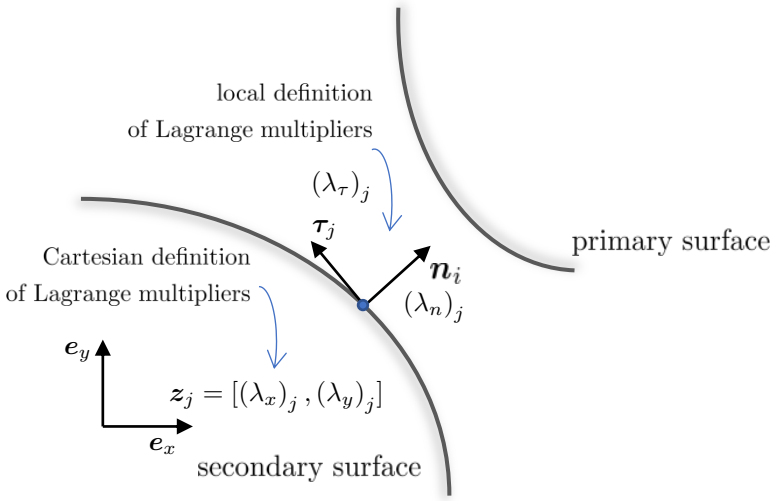


Figure 11: Simplified representation of the definition of Lagrange multipliers for imposing mechanical contact constraints. The existing definition of Lagrange multipliers using the local frame assumes that Lagrange multipliers act along the local (nodal) normal and tangent vectors. In other words, one Lagrange multiplier is introduced in the system per contact constraint equation needed. On the other hand, employing Cartesian Lagrange multipliers requires imposing two contact constraints in two dimensions and three contact constraints in three dimensions. Tangential contact constraints may impose null tangential traction (frictionless) or a Coulomb law with a friction coefficient.

5.3.2 Numerical Results

To verify the Cartesian Lagrange multipliers (see Equation (11)) and the associated new contact constraint implementations, we compare the normal Lagrange multiplier's value (λ_n) along the contact interface between the original implementation (local directions) and the new implementation (global Cartesian directions) for a two-block contact case. Specifically, we use $p_n = \boldsymbol{\lambda}_j \cdot \boldsymbol{n}_j$ to obtain the normal component of the contact traction from the new definition, and demonstrate the values for different contact interface orientations.

Figure 12 shows the normal contact pressure value along the contact interface at different refinement levels. As the refinement level increases, the curves tend to become closer to the highest refinement case, which indicates that the new implementation converges to the reference solution.

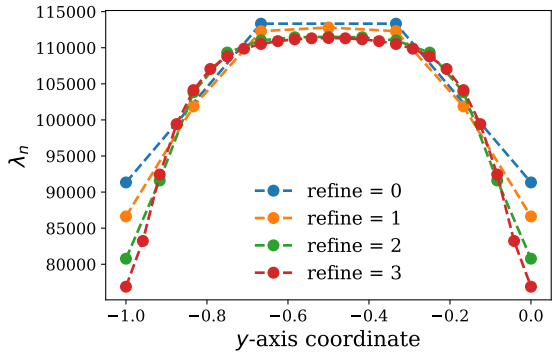


Figure 12: Normal contact pressure value along the contact interface at different mesh refinement levels.

Figure 13 shows the normal contact pressure along the contact interface at different contact interface angles with mesh refinement level equals 3. The curves for different cases are on top of each other, indicating that the contact interface angle does not affect the Lagrange multiplier's value, which is as expected. We demonstrate this case because it is found that the orientation of the contact interface has an impact on the linear solver's performance.

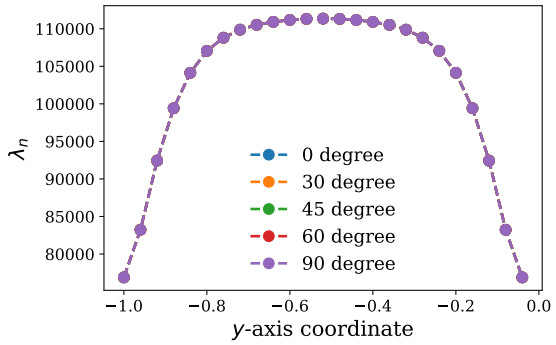


Figure 13: Normal contact pressure along the contact interface at different contact interface angles (curves are indistinguishable).

In Figure 14, we compare the normal contact pressure between the definition in the Cartesian coordinate system and that in the local coordinate system with mesh refinement level equals 3. It can be seen in Figure 14 that both formulations result in the same contact pressure. The relative difference of the two formulations is shown in Figure 14b, which indicates a maximum relative difference of 0.013%. It is noticeable that the maximum relative difference happens at the edge of the contact interface.

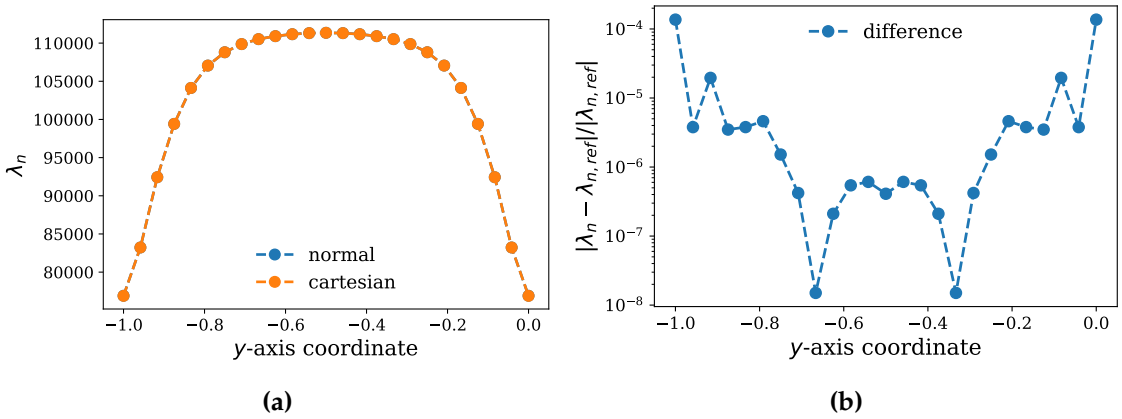
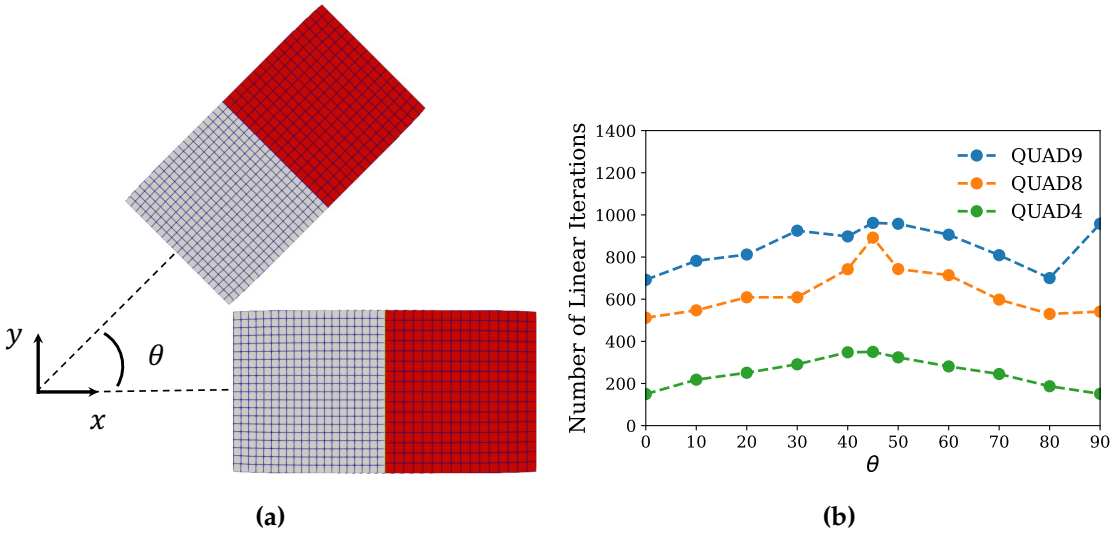


Figure 14: Comparison of the normal contact pressure distribution between the definition in the Cartesian coordinate system and that in the local coordinate system.

Figure 15 demonstrates the performance of VCP with the new Cartesian Lagrange multiplier implementation. Specifically, we look at the number of linear iterations of boomerAMG under VCP for different contact interface angles and different element types. Figure 15a shows the problem setting and the contact interface angle (θ). Here, we are solving a 2D problem with frictionless mechanical contact. Figure 15b shows the linear iteration count as a function of θ for different element types. It can be seen that element type affects the linear iteration count. At a certain θ value, QUAD9 elements require the highest linear iteration count, followed by the QUAD8 element, and the QUAD4 element. Moreover, for each element type, the θ value plays a role in the boomerAMG [17] performance as well. Specifically, the performance of boomerAMG becomes worse when θ approaches 45° (except for the QUAD9 element at $\theta = 90°$). Despite this effect, it is worth mentioning that, with the previous local definition of the Lagrange multipliers, the problem could not be solved for $\theta > 30°$. The similar phenomenon was observed and discussed in [15, 16], where convergence could not be achieved without fundamental changes in the aggregation strategy in the algebraic multigrid solver. In our work, convergence for all θ values is achieved by automatically permuting the rows of the Jacobian matrix based on the dependency of the constraint and the DoF. To enable convergence rate that is independent of θ, further investigation into the geometry-preserving coarsening strategy inside of the MOOSE framework is needed.

Despite the improvements in condensation of Lagrange multipliers employing the VCP, the mere condensation of Lagrange multipliers in the system does not enable general usage of iterative solvers in mechanical contact problems. Further steps are proposed in Section 7.



(a) **(b)**

Figure 15: Performance of boomerAMG under VCP for different contact interface angles. (a) Problem setting and illustration of the contact interface angle θ. (b) Linear iteration count as a function of contact interface angle for different element types.

6 Demonstration of Thermomechanical Mortar in BISON Cases

This section describes BISON applications of thermomechanical mortar in 2D and 3D cases. These applications leverage the extensions of physics to 3D detailed in Section 3 and Section 4.

6.1 Comparison of node-to-segment (NTS) to Mortar in 2D Cases

This section shows 2D cases added to the BISON repository, which use the mortar approach to thermomechanical interface problems.

6.1.1 TRISO Problem

Usage of thermomechanical mortar is extended to non-LWR applications. Figure 16 shows mechanical contact pressure and temperature results for a two-dimensional model of a TRISO particle, which is modeled using axisymmetry (on the left side), in an accident scenario. The layers initially make mechanical contact and quickly separate; after that, only thermal contact takes place on the open interface shown in the figure. While a penalty enforcement of node-to-segment contact typically facilitates numerical convergence, a poor choice of parameters can affect normal contact pressure distribution (see image at the top right of Fig. 16). Exact enforcement of mortar mechanical contact constraints eliminates these numerical edge-related artifacts. For problems in which edge-to-edge mechanical contact is expected, users can set the option `correct_edge_dropping = true` to ensure that projections from the secondary surface to the primary surfaces are correctly resolved. This option can result in poor scaling of the system equations, which can be alleviated by condensing the Lagrange multipliers (i.e. usage of the variable condensation preconditioner (VCP)).

6.1.2 Calvert Cliffs: UFE 019

LWR models in which elongation is a key metric require the consideration of friction on the fuel-cladding interface. Often these models are run without friction or using "glued" contact. Whereas frictionless contact models cannot capture the sticking behavior of fuel-cladding interface, "glued" approaches neglect slipping behavior that may occur due to the (local) flexibility of the fuel rod. The latter physical phenomenon is more prevalent when pellets are modeled as discrete (chamfer and dish), since each pellet contact interface may have partial stick and slip regions, whose contribution to the overall fuel rod elongation would not be correctly represented as frictionless or "glued".

Here, we take an existing BISON Calvert Cliffs UFE019 model that leverages an augmented Lagrange approach to solve the frictional problem. Both mortar- and node-to-segment-based input files are available in the BISON repository. Figures 17 and 18 show performance metrics of mortar and NTS formulations for a 2D-RZ frictional LWR problem. The augmented Lagrange approach employed by the NTS formulation requires additional outer loops to arrive at the contact solution, which is determined by a set of tolerances. The mortar approach uses a semi-smooth Newton approach with three branches defining nodal states: In contact and slipping, in contact and sticking, and not in contact. The semi-smooth Newton approach to arrive at the frictional contact solution shows numerical advantages in often-run 2D-RZ LWR simulations. When mechanical contact is established or significant fuel-cladding contact pressure changes take place, mortar contact can arrive at the frictional solution within solver tolerance in fewer number of Newton iterations compared to the existing approach.

Figures 19 and 20 show that both frictional formulations arrive at roughly the same fuel and rod elongation results. In Figure 20, we mark with black, green, and yellow stars the final elongation reading for the frictional and frictionless simulations, and experimental data point,

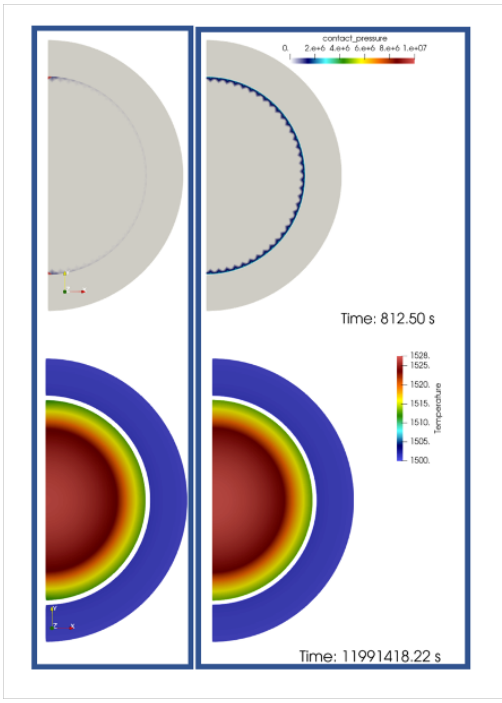


Figure 16: Tri-structural isotropic particle fuel two-dimensional simulation in BISON. On the left, node-to-segment results of contact pressure and temperature (note the uneven contact pressure distribution). On the right, the same model run with exact thermomechanical mortar enforcement with Lagrange multipliers. There results are shown at two stages of the simulation.

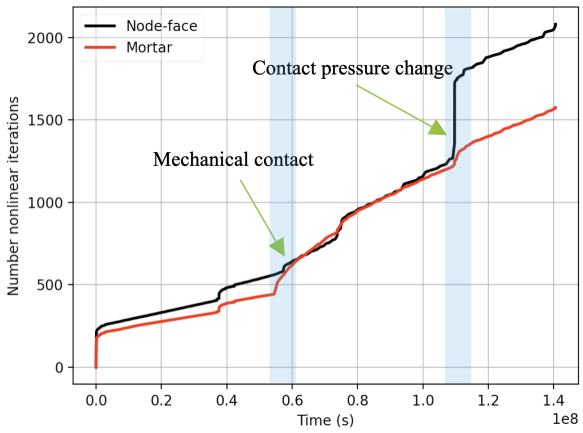


Figure 17: Cumulative number of nonlinear (Newton) iterations (NTS vs mortar).

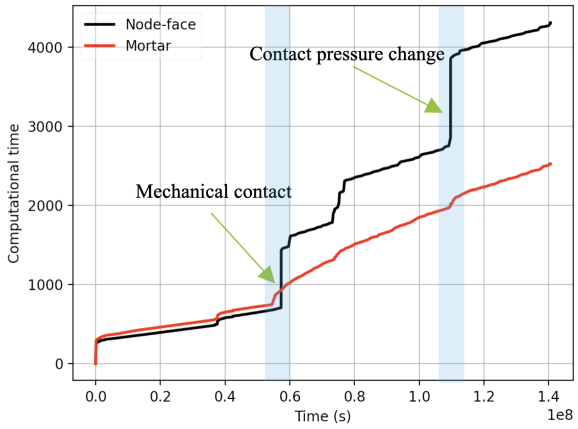


Figure 18: Wall time in seconds (NTS vs mortar).

respectively. It is apparent that the contact formulation used to capture friction has no effect on the elongation results. In the case presented in Figure 20, a coefficient of friction of 0.4 was used. It is expected that, with lower friction, the mechanical coupling between fuel and cladding be delayed and, consequently, the frictional results of elongation (33.2 mm) get closer to the measured data (28.3 mm).

As expected, the action of friction significantly affects the axial motion of the fuel rod assembly,

while not having much effect on the pellet stack's axial motion.

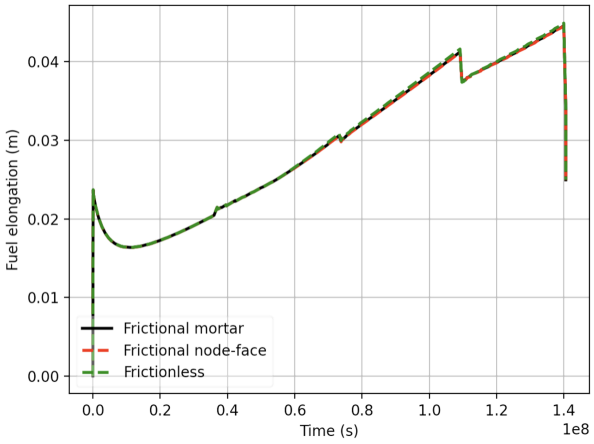


Figure 19: Fuel elongation history for Calvert Cliffs - UFE019. Mortar and NTS frictional solutions are compared to frictionless results.

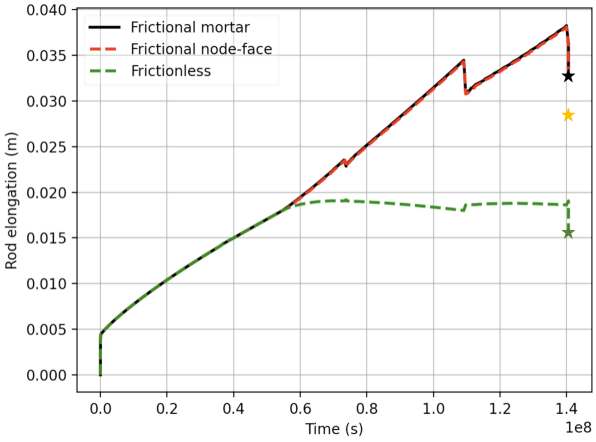


Figure 20: Cladding (rod) elongation history for Calvert Cliffs - UFE019. Final mortar and NTS frictional solutions (black star) are compared to frictionless results (green star). The available elongation data point obtained experimentally is marked with a yellow star.

6.2 3D Fuel Rod Example

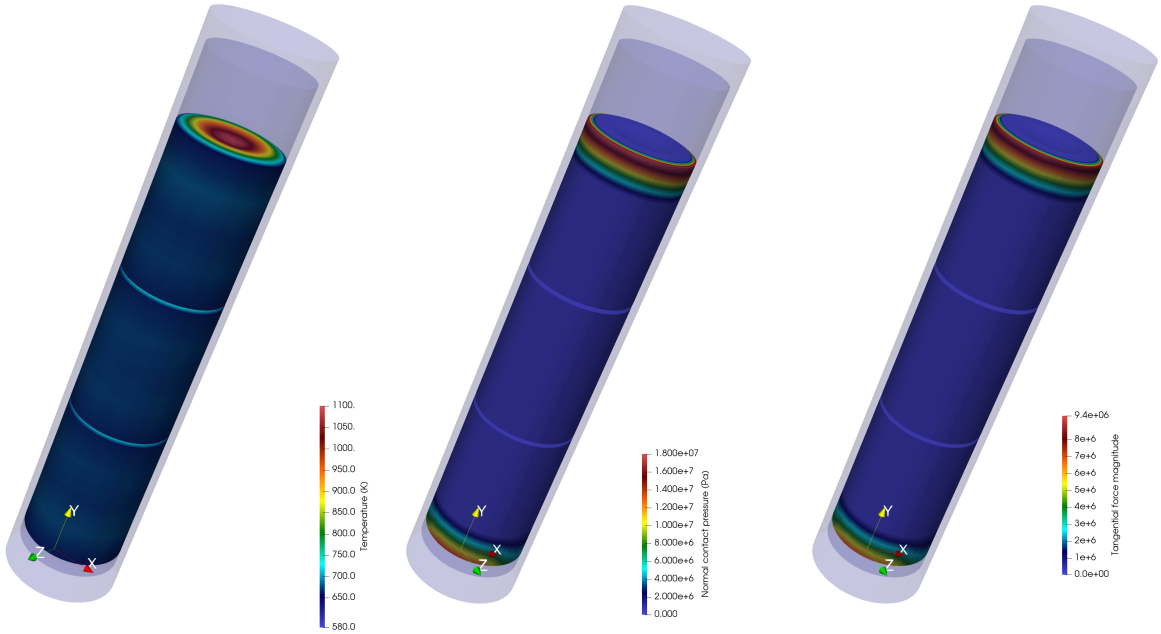
This section demonstrates the application of the mortar finite element method to a general three-dimensional LWR thermomechanical problem. While three-dimensional LWR models are expensive to simulate, they represent a modeling opportunity to include manufacturing defects and experimental setup inaccuracies, in addition to physics-motivated asymmetries in the problem, such as cracking or fragmentation.

Gap heat transfer physics for LWR-type of problems, developed last fiscal year [18], have been extended here to three-dimensional pellet-rod problems. The model consists of three aligned discrete pellets in a cladding encase with typical LWR physics, including heat conduction, neutron heat source, plenum pressure, coolant channel pressure, burnup, swelling, thermal expansion, and creep, with UO_2 and Zircaloy-4 material properties. While the material model details are not provided in this document, they are available in the BISON repository under the `examples` directory.

The selection of boundary conditions in three-dimensions requires special care. Cylindrical symmetry can be easily applied to the pellets by restricting the motion of the pellets at their centerline. In addition, pellet rotation needs to be prevented, which is done by selecting axis-aligned nodes and constraining its motion to lie on radial/axial plane. The bottom of the cladding is fixed, which avoids its rigid body motion and reduces the strain level at the bottom of the short rod. A contiguous mesh is used between pellets, so no additional boundary conditions on the pellet-pellet interface are required.

While this setup is not fully representative of LWR fuel rods (e.g. elongation metrics would not match those of a full rod), conclusions can be drawn as to the mechanical behavior of the fuel rod during the reactor's life cycle. A coefficient of friction of 0.5 causes the volumetric expansion of fuel to progressively achieve a state of overall stick on the contact interface, which couples

the axial displacement of fuel and cladding materials. Figures 21a to 21c depict temperature and contact pressures distribution at an intermediate stage of the simulation; it can be observed that contact pressure accumulate at the edges of the pellets. If the pellet-cladding interface is frictionless, cladding remains unaffected by the fuel's longitudinal motion. Figure 22 provides additional details.



(a) **(b)** **(c)**

Figure 21: (a) Temperature distribution on 3D pellet example. (b) Normal contact pressure distribution on 3D pellet example. (c) Frictional contact pressure magnitude distribution on 3D pellet example.

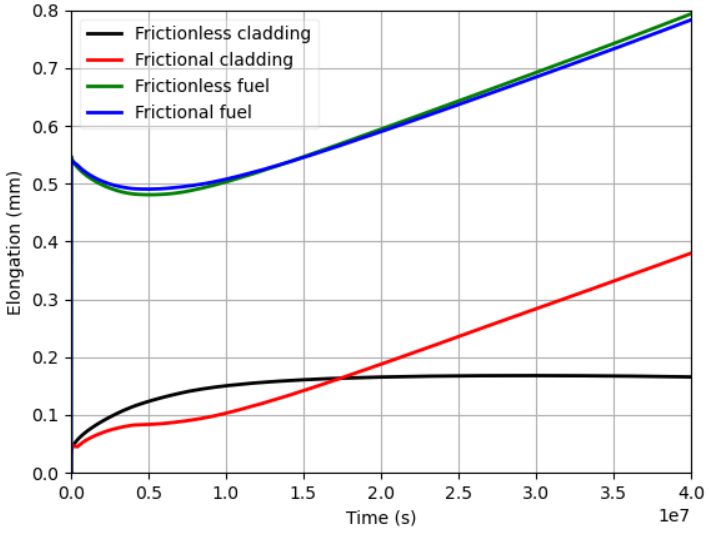


Figure 22: Centerline longitudinal displacement at the top of fuel and cladding. Mechanical contact in this simulation takes place at the beginning ($\approxeq$ 3 hours), at which time, mechanical coupling between fuel and cladding starts. If friction is present (red and blue lines), as the contact surface area between fuel and cladding increases, the fuel rod elongation evolution resembles that of the fuel. At about four months into the simulation, friction generates an overall stick state between fuel and cladding, which causes the cladding (red line) to follow the behavior of the fuel (blue line). If no friction is present, while the fuel elongation is almost unaffected (other than a small additional resistance to axial motion), cladding elongation stays virtually constant for the last three quarters of simulation.

This three-pellet example with friction was added to the BISON repository and documented.

6.3 Three-Dimensional Fuel Rod Example with Eccentricity and Defect

In the previous section, results of symmetric pellet geometry were shown. However, when the model exhibits cylindrical symmetry, three-dimensional models typically become too expensive to run and two-dimensional models with axisymmetry ("2D-RZ") are employed instead. There are scenarios in which these simplifications no longer hold. For example, practical reactor setups often include manufacturing or construction defects, which may affect the geometry and eccentricity of fuel pins. These defects have the potential to affect the temperature and stress fields due to varying gap distances and the irregular distribution of normal and tangential contact pressure.

In this section, we simulate a three-dimensional five pellet example extracted from a BWR rod. The pellets are meshed individually with a given eccentricity. The eccentricities are chosen to be 15 μm, -25 μm, 0 μm, 35 μm, and -35 μm for the five pellets, from bottom to top, along the global X axis. The solution to the two-dimensional axisymmetric problem is used to impose local conditions in the three-dimensional problem. Namely, fission gas release, gas temperature in the plenum, and plenum pressure from the global model are applied to the local, three-dimensional model. Heat conduction, neutron heat source, gap heat transfer, creep, and thermal expansion are considered in this simulation, which includes specific properties of zircaloy-4 and UO_2. Details on the parameters, other than eccentricity, employed in this BISON simulation can be consulted in [12]. That work is here extended to consider frictional mechanical interaction between fuel and cladding. In addition, pellets are linked to each other via mortar displacement continuity boundary conditions, which prevents overall rigid body motion. The solution to this system is

obtained using a Newton scheme, as opposed to the more often used preconditioner Jacobian-free Newton Krylov (PJFNK) strategy.

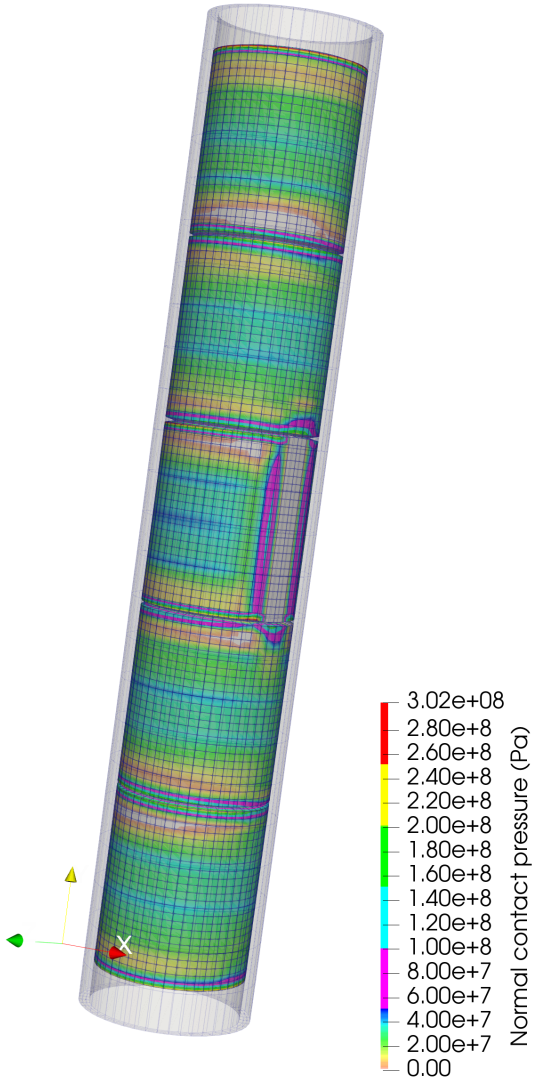


Figure 23: Normal contact pressure distribution on missing pellet surface model.

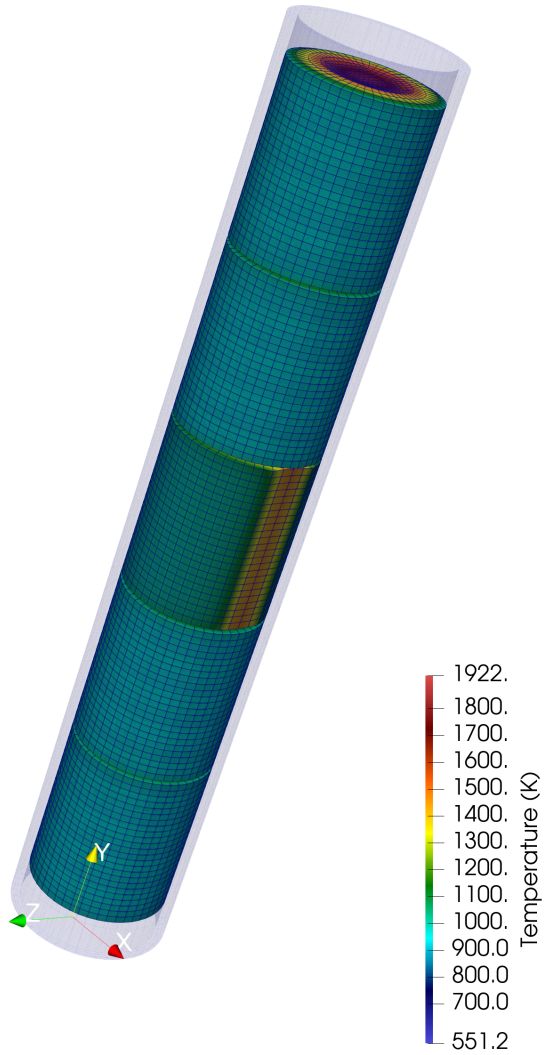


Figure 24: Temperature distribution on missing pellet surface model.

The enforced eccentricity of pellets with respect to the rod centerline creates an uneven distribution of contact pressure as the pellet-cladding gap varies longitudinally and radially as the pellets expand and gradually come into contact with the cladding material (see Fig. 23). The effects of lack of cooling on the missing pellet surface area is observed in Fig. 24.

Figures 25 and 26 show the evolution of elongation and the relative axial motion between fuel and cladding at the top of the fuel rod. Despite existing areas on which no contact occurs between fuel and pellet (i.e. chamfer and fuel rod end regions), friction on the contact interface stabilizes the overall relative displacement at about 0.3 mm. On the other hand, the longitudinal expansion of fuel does not affect the axial deformation of the cladding if no friction is present.

Stresses in Pa along the longitudinal direction are depicted in Figures 27 and 28, for a frictionless

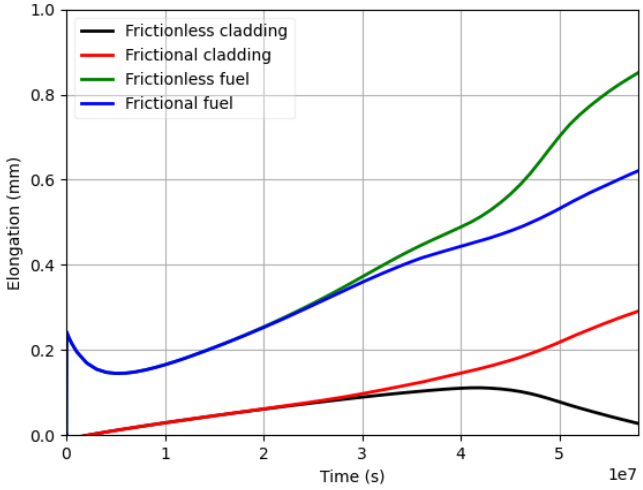


Figure 25: Evolution of axial elongation at the top of fuel and cladding for frictionless and frictional behavior. Elongation is measured at the top of the fuel stack and at the top of cladding section.

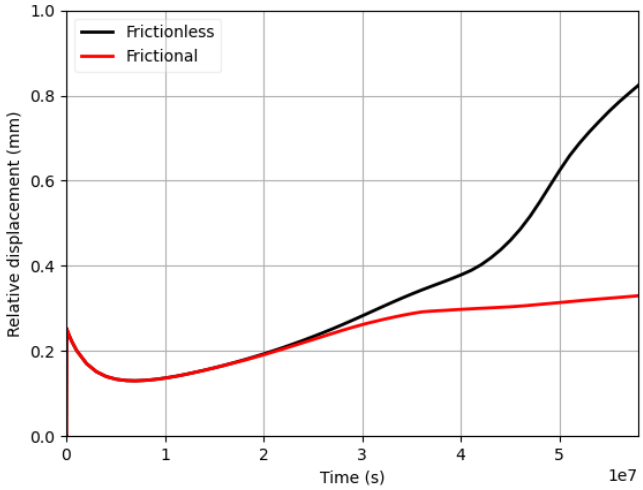


Figure 26: Differential elongation at the top of the fuel rod segment with and without friction. It is observed that, to a great extent, the presence of friction links the axial motion of fuel and rod. Since elongation is measured at a point where no direct fuel-cladding contact occurs, the deformation of cladding due to creep and thermal expansion contributes to the relative axial motion.

and a frictional case, respectively. Differences are noticeable: While the frictionless case shows low values of compression stresses, the frictional case yields axial stresses on the order of 200 MPa-250 MPa in the cladding. The compressing action of the coolant along the axial direction[2] was not considered in this setup, which would have an alleviating effect on the stress values in Figure 28.

[2]In essence, the effect of the difference between the plenum pressure and the coolant pressure at both ends of the fuel rod as it create compression forces on the cladding perimeter

As expected, the influence of friction on the thermal problem is not shown to be significant (see Figures 29 and 30).

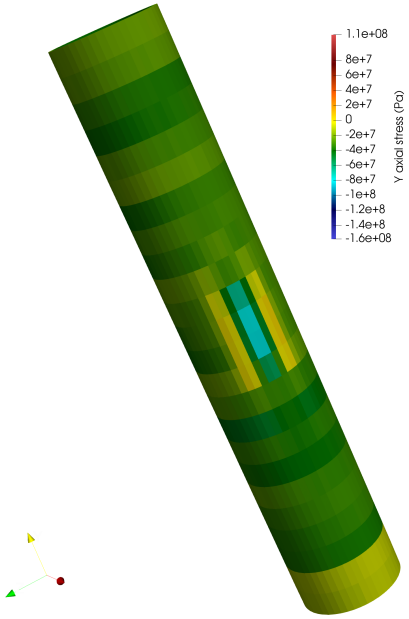


Figure 27: Axial stress on cladding for a frictionless simulation.

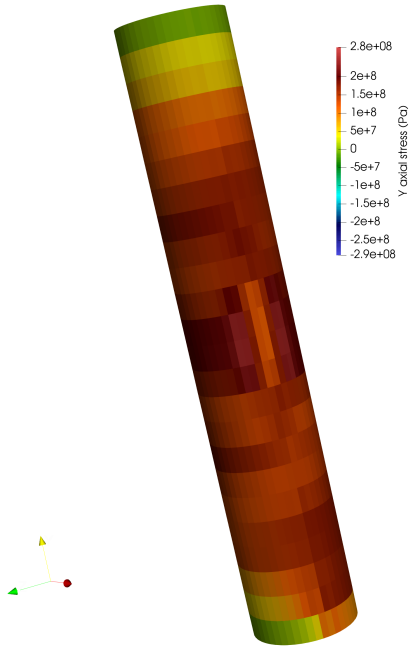


Figure 28: Axial stress on cladding for a frictional simulation ($\mu = 0.5$).

Figure 29: Temperature on cladding for a frictionless simulation.

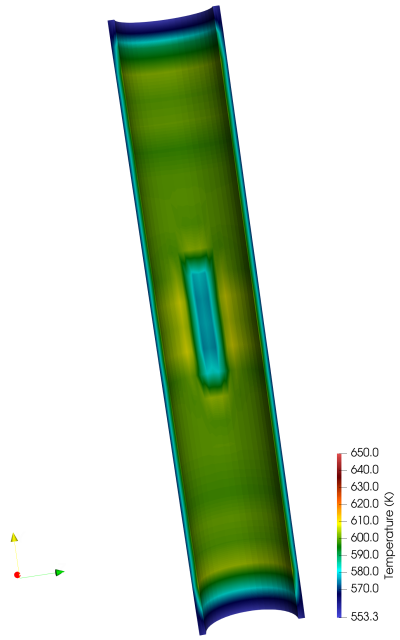


Figure 30: Temperature on cladding for a frictional simulation ($\mu = 0.5$).

The effect of pellet eccentricity on cladding deformation may be observed in Figure 31, where displacements are exaggerated. Similarly, the missing pellet surface defect provides an opportunity for the cladding to creep in as the simulation progresses at relatively high temperature (see Figure 32).

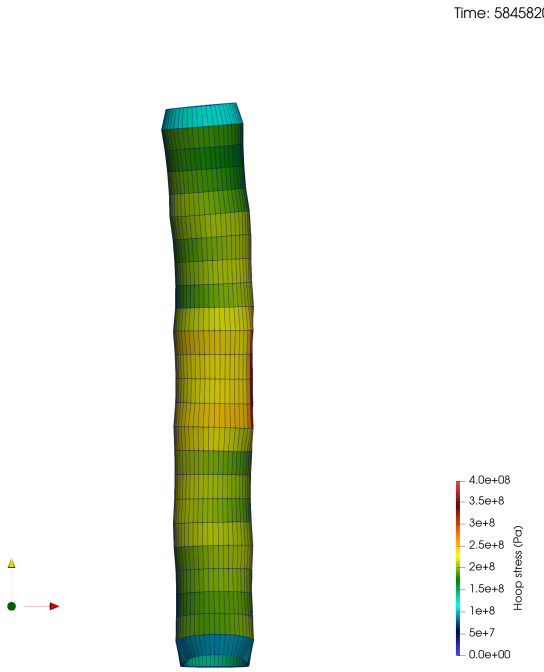


Figure 31: Cladding hoop stress distribution as viewed on the XY plane (pellet eccentricity can be observed). Displacements are augmented by a factor of 50.

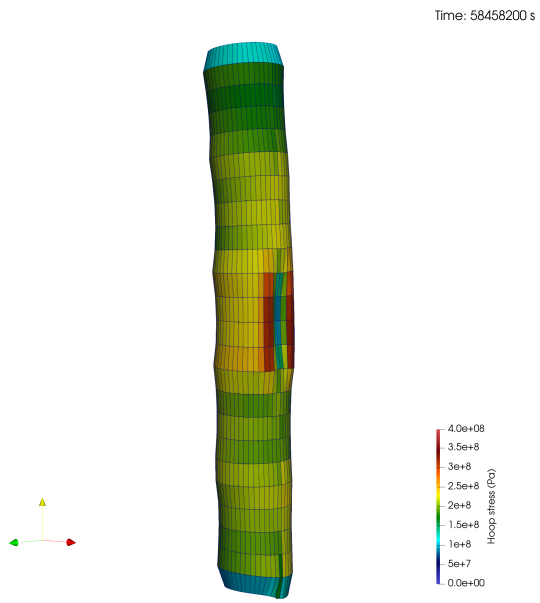


Figure 32: Cladding hoop stress distribution as viewed from an oblique perspective. Creeping of the cladding into the missing pellet surface area can be observed. Displacements are augmented by a factor of 50.

6.4 A Note on the Problem Setup

BISON's thermomechanical LWR models with a mortar approach and second-order Lagrange meshes typically require increasing c_n and c_t coefficients for mechanical contact (see Equation (4)), which is due to the influence of element size, shape functions, and contact pressure values in the contact constraint equations, which are evaluated at quadrature points in the mortar segment mesh. For large, 3-D practical BISON problems, it is observed that problem convergence is more dependent on the selection of a narrow range of values for c_n and c_t. This phenomenon, which does not happen in plain solid mechanics problems, is attributable to worsening of the system's conditioning and can be alleviated by manually scaling the normal and tangential contact Lagrange multiplier and the heat flux Lagrange multiplier vectors, such that their values come closer to unity.

7 Conclusions and Future Work

A summary of goals achieved in this work is provided in Table 1. Tasks performed this fiscal year leveraged the creation of mortar segment meshes for 3D problems and accordingly extended gap heat transfer and contact mechanics physics to work with generality in three dimensions. These developments were applied to BISON three dimensional LWR problems, including friction on the pellet-cladding contact interface. In 2D, mortar contact mechanics shows convergence improvements, especially when friction is employed. Of more relevance is the possibility of including friction with exact enforcement in 3D BISON problems, which was enabled by the mortar framework. This allows for numerically exploring manufacturing defects and setup misalignment in nuclear reactor models without compromising accuracy in the solid mechanics solution. Extension of mortar constraints to using a penalty approach will relax some numerical difficulties such as the saddle point problem generated by the presence of Lagrange multipliers and the need for considering those additional variables.

The VCP was extended to work with generality by condensing Lagrange multipliers (modeling contact pressure) defined in the global Cartesian coordinate system. This condensation helps avoid some numerical issues, such as scaling (system's conditioning) problems that appear when secondary surfaces do not fully project onto the primary surface, and represents a key step towards the use of iterative solvers in contact problems, i.e. improving scalability of mechanical problems with contact. While the VCP and Cartesian Lagrange multipliers can be employed with iterative solvers, the resulting performance is suboptimal and does not allow its use in practical industrial problems. It is known that contact interfaces need to be treated especially when dealing with iterative solvers (see, e.g., [15, 16]); strategies to account for the contact interface in the context of the MOOSE-PETSc [2] interface will be investigated.

Other items for future work are described in Section 1.2.

Table 1: Summary of Developments and Outcomes

Modeling Aspect	**Result**
Gap heat transfer extension to 3D	A modular approach to accounting for gap heat transfer physics using mortar was developed in MOOSE and the existing LWR thermal contact constraint object was extended to work with 3D simulations (see Section 3). Even simple problems show improved convergence behavior over the NTS solution (see Figure 4).
Mechanical contact extension to 3D	The frictional formulation was extended to three dimensions. Results of a benchmark problem may be consulted in Figures 5 and 6.
Variable condensation preconditioner	The computation of dual shape functions were extended to work with cases in which the secondary surface does not fully project onto the primary surface (see Figure 8). Usage of the VCP drastically improves system conditioning in edge dropping cases (see Figure 10). Its use was generalized for contact happening in oblique directions by its joint use with Cartesian Lagrange multipliers (see Section 5.3.1).
3D mortar in BISON	Thermomechanical mortar was applied to 2D and 3D LWR cases. Friction is shown to yield similar elongation results compared to the older NTS formulation (see Figure 20) while contributing to more realistic fuel rod elongation results. The approach developed here allowed for running frictional pellet-cladding interfaces in 3D short stacks where pellet suffered from eccentricities and defects (see Figures 23, 25 and 32).

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 report, or allow others to do so, for U.S. Government purposes.

This research made use of the resources of the High Performance Computing Center at Idaho National Laboratory, which is supported by the DOE Office of Nuclear Energy and the Nuclear Science User Facilities under contract no. DE-AC07-05ID14517.

References

[1] Mortar migration guide, Sep 2022. URL http://mooseframework.inl.gov/bison/tutorials/mortar_conversion/mortar_conversion_guidelines.html.

[2] Satish Balay, Shrirang Abhyankar, Mark Adams, Jed Brown, Peter Brune, Kris Buschelman, Lisandro Dalcin, Alp Dener, Victor Eijkhout, W Gropp, et al. Petsc users manual. 2019.

[3] T Cichosz and M Bischoff. Consistent treatment of boundaries with mortar contact formulations using dual lagrange multipliers. *Computer Methods in Applied Mechanics and Engineering*, 200(9-12):1317–1332, 2011.

[4] Stefan Hüeber, Georg Stadler, and Barbara I Wohlmuth. A primal-dual active set algorithm for three-dimensional contact problems with coulomb friction. *SIAM Journal on scientific computing*, 30(2):572–596, 2008.

[5] Benjamin S. Kirk, John W. Peterson, Roy H. Stogner, and Graham F. Carey. libMesh: A C++ Library for Parallel Adaptive Mesh Refinement/Coarsening Simulations. *Engineering with Computers*, 22(3–4):237–254, 2006. https://doi.org/10.1007/s00366-006-0049-3.

[6] Daniel S Lopes, Mauro T Silva, and Jorge A Ambrósio. Tangent vectors to a 3-d surface normal: A geometric tool to find orthogonal vectors based on the householder transformation. *Computer-Aided Design*, 45(3):683–694, 2013.

[7] Panayiotis Papadopoulos and Robert L Taylor. A mixed formulation for the finite element solution of contact problems. *Computer Methods in Applied Mechanics and Engineering*, 94(3): 373–389, 1992.

[8] Cody J Permann, Derek R Gaston, David Andrš, Robert W Carlsen, Fande Kong, Alexander D Lindsay, Jason M Miller, John W Peterson, Andrew E Slaughter, Roy H Stogner, et al. Moose: Enabling massively parallel multiphysics simulation. *SoftwareX*, 11:100430, 2020.

[9] Alexander Popp and WA Wall. Dual mortar methods for computational contact mechanics–overview and recent developments. *GAMM-Mitteilungen*, 37(1):66–84, 2014.

[10] Alexander Popp, Alexander Seitz, Michael W Gee, and Wolfgang A Wall. Improved robustness and consistency of 3d contact algorithms based on a dual mortar approach. *Computer Methods in Applied Mechanics and Engineering*, 264:67–80, 2013.

[11] Antonio Recuero, Alexander Lindsay, Dewen Yushu, John W. Peterson, and Benjamin Spencer. A mortar thermomechanical contact computational framework for nuclear fuel performance simulation. *Nuclear Engineering and Design*, 394:111808, 2022. ISSN 0029-5493. doi: https://doi.org/10.1016/j.nucengdes.2022.111808. URL https://www.sciencedirect.com/science/article/pii/S0029549322001625.

[12] B.W. Spencer, R.L. Williamson, D.S. Stafford, S.R. Novascone, J.D. Hales, and G. Pastore. 3d modeling of missing pellet surface defects in bwr fuel. *Nuclear Engineering and Design*, 307: 155–171, 2016. ISSN 0029-5493. doi: https://doi.org/10.1016/j.nucengdes.2016.07.008. URL https://www.sciencedirect.com/science/article/pii/S0029549316302187.

[13] Aysenur Toptan, David J. Kropaczek, and Maria N. Avramova. Gap conductance modeling ii: Optimized model for uo2-zircaloy interfaces. *Nuclear Engineering and Design*, 355:110289,

2019. ISSN 0029-5493. doi: https://doi.org/10.1016/j.nucengdes.2019.110289. URL https://www.sciencedirect.com/science/article/pii/S0029549319303061.

[14] Aysenur Toptan, Jason D. Hales, Richard L. Williamson, Stephen R. Novascone, Giovanni Pastore, and David J. Kropaczek. Modeling of gap conductance for LWR fuel rods applied in the BISON code. *Journal of Nuclear Science and Technology*, 57(8):963–974, 2020. ISSN 0022-3131. doi: 10.1080/00223131.2020.1740808.

[15] TA Wiesner, A Popp, MW Gee, and WA Wall. Algebraic multigrid methods for dual mortar finite element formulations in contact mechanics. *International Journal for Numerical Methods in Engineering*, 114(4):399–430, 2018.

[16] Tobias A Wiesner, Matthias Mayr, Alexander Popp, Michael W Gee, and Wolfgang A Wall. Algebraic multigrid methods for saddle point systems arising from mortar contact formulations. *International Journal for Numerical Methods in Engineering*, 122(15):3749–3779, 2021.

[17] Ulrike Meier Yang et al. Boomeramg: A parallel algebraic multigrid solver and preconditioner. *Applied Numerical Mathematics*, 41(1):155–177, 2002.

[18] Dewen Yushu, Antonio Martin Recuero, Daniel Schwen, Alexander D Lindsay, and Benjamin W Spencer. M3 milestone: Advanced contact 2021. Technical report, Idaho National Lab.(INL), Idaho Falls, ID (United States), 9 2021. URL https://www.osti.gov/biblio/1836106.