INL/EXT-17-43399

BISON Contact Algorithm Improvements in Support of Pellet Cladding Mechanical Interaction Modeling

B. W. Spencer, J. W. Peterson, W. Jiang,
Y. Liu, S. Veeraraghavan, A. Casagranda

September 2017

The INL is a U.S. Department of Energy National Laboratory
operated by Battelle Energy Alliance

INL/EXT-17-43399

BISON Contact Algorithm Improvements in Support of Pellet Cladding Mechanical Interaction Modeling

B. W. Spencer, J. W. Peterson, W. Jiang, Y. Liu, S. Veeraraghavan, A. Casagranda

September 2017

Idaho National Laboratory
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

CASL-U-2017-1430-000
INL/EXT-17-43399

BISON Contact Algorithm Improvements in Support of Pellet Cladding Mechanical Interaction Modeling

L2:FMC.P15.10

B. W. Spencer, INL
J. W. Peterson, INL
W. Jiang, INL
Y. Liu*, INL
S. Veeraraghavan, INL
A. Casagranda, INL

September 29, 2017

*Currently at Duke University

REVISION LOG

Revision	Date	Affected Pages	Revision Description
0000	9/29/2017	All	Initial Release

Document pages that are:

Export Controlled _____

IP/Proprietary/NDA Controlled_____

Sensitive Controlled_____

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

Requested Distribution:

To:

Copy:

BISON Contact Algorithm Improvements in Support of Pellet Cladding Mechanical Interaction Modeling

B. W. Spencer, J. W. Peterson, W. Jiang, Y. Liu, S. Veeraraghavan, A. Casagranda
Fuels Modeling and Simulation
Idaho National Laboratory
P.O. Box 1625
Idaho Falls, ID 83415-3840

September 29, 2017

Introduction

The thermal and mechanical behavior of the gap between the fuel and cladding plays an extremely important role in determining the response of light water reactor (LWR) fuel under irradiation. This is especially true during situations when pellet-cladding mechanical interactions (PCMI) have a significant effect on the behavior of the fuel/cladding system. The BISON code, which is being used as the fuel performance simulation tool for the CASL program, employs contact detection and enforcement algorithms to realistically simulate the thermal and mechanical interactions across the fuel/cladding gap in LWR fuel. These contact algorithms permit accurate representation of these interactions even when there is significant relative movement between the two surfaces, which is typically the case in many regions of an LWR fuel rod.

This report summarizes work done during the 2017 fiscal year to further improve the accuracy and robustness of the contact enforcement algorithms in BISON. This work will benefit all BISON simulations, but will particularly benefit simulations scenarios where PCMI is of interest. Work was done in the following areas:

- **Mortar Method for Thermal Contact Enforcement** Mechanical and thermal contact are currently enforced in BISON using node/face contact constraints. While node/face contact is relatively straightforward to implement, it suffers from a number of accuracy and stability issues. During this fiscal year, the first steps were taken to implement a mortar algorithm that enforces contact using face/face constraints. This method has the potential to significantly improve the mechanical and thermal solutions, especially in the vicinity of the pellet/cladding interface, which will lead to improved accuracy in PCMI simulations. It is expected to also overcome issues with nonphysical spatial oscillations in the thermal and mechanical solutions along the length of LWR fuel rods. The current work has focused on using this method for enforcement of thermal contact in 2D models, but this approach can be extended to work in 3D and for mechanical contact enforcement.
- **Augmented Lagrange Multiplier Method for Mechanical Contact Enforcement** BISON currently offers two main options for the method used to enforce mechanical contact constraints: penalty and kinematic. The penalty method is the simplest approach, representing contact interactions as springs with a user-defined stiffness. The drawback of the penalty approach is that when contact occurs, forces must be accompanied by penetration, which can be significant depending on the penalty factor chosen. The kinematic approach avoids this issue, but can suffer from convergence robustness issues, especially for frictional contact. The Augmented Lagrange Method (ALM) performs a series of solutions in which a penalty approach is used to enforce contact interactions during the nonlinear iterations. Once the solution is obtained, it is modified to remove those nonphysical penetrations. This process is iteratively repeated until there is no longer a need to modify the solution. This approach is expected to improve the robustness of frictional mechanical contact enforcement.

- **Improvements to Contact Search Robustness** In BISON's contact search algorithm, a set of faces on the master surface nearest to a slave node is stored at the beginning of either a time step or the entire analysis, depending on the options used. The contact search only considers interactions of the slave node with this set of faces (known as the contact patch). This is primarily done for efficiency, as it greatly reduces the cost of the search performed at each iteration, and allows for pre-allocation of off-diagonal entries in the preconditioning matrix for contact. However, if a node slides out of that original patch during the nonlinear iterations within a step, there was no provision in the algorithm to enforce that constraint. So if an insufficiently large patch was used, situations could arise when contact would not be enforced on some nodes when it should have been. To address this issue, the contact search has been modified to better handle those cases, to ensure that contact is always enforced when it should be.

The details of the developments in these areas are described in corresponding sections of this report.

1 Mortar Method for Thermal Contact Enforcement

The mortar finite element method has been used for many years in a variety of applications, including the enforcement of continuity conditions across decomposed domains [1], the implementation of Dirichlet boundary conditions [2–4], obtaining improved estimates of surface fluxes [5], and for solving large deformation contact mechanics problems [6–11]. There is a currently great deal of interest in developing more robust mechanical and thermal contact solution strategies in the BISON fuel performance simulation application, and schemes based on the mortar finite element approach appear to be a promising avenue of development. There are a number of challenges associated with the development of a robust Lagrange multiplier based formulation of the mortar finite element method which can be tackled using the relatively simple framework of thermal contact problems, before moving on to the "real" application of thermomechanical contact. In this section, we describe several aspects of our mortar finite element method implementation for solving the thermal contact (also known as the gap heat conduction) problem, which is based on the work of Yang et al. [12, 13].

1.1 Heat conduction in multiple bodies

The following description closely follows the formulation of Gitterle [14], and is also related to methods described in other works [15, 16]. We consider the problem of computing the temperature distribution in two regions $\Omega^{(1)}$, $\Omega^{(2)}$ with scalar thermal conductivities $k^{(1)}$, $k^{(2)}$, which are either in contact or separated by a small gap. We shall use parenthetical superscript indices throughout to refer to the region on which a particular quantity is defined.

The temperature in each region satisfies the heat conduction equation

$$-\nabla \cdot k^{(m)} \nabla T^{(m)} = f^{(m)} \quad \in \Omega^{(m)} \tag{1}$$

for $m = 1, 2$, where $f^{(m)}$ is a specified heat source in body m. The regions are in contact (or near-contact) on a shared segment of their boundaries denoted by $\Gamma_C^{(1)}$ and $\Gamma_C^{(2)}$, respectively. We assume that Dirichlet and Neumann boundary conditions for body m are specified on $\Gamma_D^{(m)}$ and $\Gamma_N^{(m)}$, respectively, i.e.

$$T^{(m)} = T_D^{(m)} \in \Gamma_D^{(m)} \tag{2}$$

$$-k^{(m)} \nabla T^{(m)} \cdot \hat{n}^{(m)} = q_N^{(m)} \in \Gamma_N^{(m)} \tag{3}$$

where $T_D^{(m)}$ and $q_N^{(m)}$ are given data, and $\hat{n}^{(m)}$ is the outward unit normal on $\Omega^{(m)}$. For simplicity, we also assume that these regions are non-overlapping:

$$\partial\Omega^{(m)} = \Gamma_D^{(m)} \cup \Gamma_N^{(m)} \cup \Gamma_C^{(m)} \tag{4}$$

$$\emptyset = \Gamma_D^{(m)} \cap \Gamma_N^{(m)} \cap \Gamma_C^{(m)} \tag{5}$$

The thermal conductivities $k^{(m)}$ may depend on both spatial location and temperature. The bodies and their different boundaries are shown schematically in Fig. 1.

We will denote the heat flux (per unit area) through the surface $\Gamma_C^{(m)}$ by $q_C^{(m)}$. If we assume there is no frictional dissipation and no energy stored in the contact interface, then the heat fluxes through $\Gamma_C^{(m)}$ can be expressed in terms of the temperatures in the respective bodies as

$$q_C^{(1)} = h(T^{(1)} - T^{(2)}) \tag{6}$$

$$q_C^{(2)} = -h(T^{(1)} - T^{(2)}) \tag{7}$$

where h is an appropriate heat transfer coefficient. Obviously, in this simple situation, we have:

$$q_C^{(1)} = -q_C^{(2)} \tag{8}$$

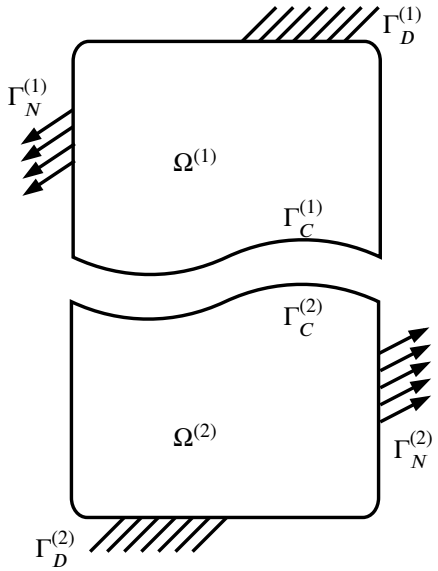


Figure 1: Boundary conditions and configuration of the bodies.

The weak formulation of (1) proceeds by first defining appropriate test and trial spaces

$$\mathcal{V}^{(m)} \equiv \{v^{(m)} \in H^1\left(\Omega^{(m)}\right) : v^{(m)} = 0 \in \Gamma_D^{(m)}\} \tag{9}$$

$$\mathcal{S}^{(m)} \equiv \{T^{(m)} \in H^1\left(\Omega^{(m)}\right) : T^{(m)} = T_D^{(m)} \in \Gamma_D^{(m)}\} \tag{10}$$

The weak formulation of the problem for body m is then: find $T^{(m)} \in \mathcal{S}^{(m)}$ such that

$$\int_{\Omega^{(m)}} \left(k^{(m)} \nabla T^{(m)} \cdot \nabla v^{(m)} - f^{(m)} v^{(m)}\right) \mathrm{d}x + \int_{\Gamma_N^{(m)}} q_N^{(m)} v^{(m)} \, \mathrm{d}s + \int_{\Gamma_C^{(m)}} q_C^{(m)} v^{(m)} \, \mathrm{d}s = 0 \tag{11}$$

holds for every $v^{(m)} \in \mathcal{V}^{(m)}$. The weak formulation for the full problem is obtained by summing the terms of (11) for both bodies, which we can write as: find $(T^{(1)}, T^{(2)}) \in \mathcal{S}^{(1)} \times \mathcal{S}^{(2)}$ such that

$$\sum_{m=1}^{2} \left[\int_{\Omega^{(m)}} \left(k^{(m)} \nabla T^{(m)} \cdot \nabla v^{(m)} - f^{(m)} v^{(m)}\right) \mathrm{d}x + \int_{\Gamma_N^{(m)}} q_N^{(m)} v^{(m)} \, \mathrm{d}s \right] + \int_{\Gamma_C^{(1)}} q_C^{(1)} v^{(1)} \, \mathrm{d}s - \int_{\Gamma_C^{(2)}} q_C^{(1)} v^{(2)} \, \mathrm{d}s = 0 \tag{12}$$

holds for all $(v^{(1)}, v^{(2)}) \in \mathcal{V}^{(1)} \times \mathcal{V}^{(2)}$. Note that, in the last line of (12), we have used (8) to eliminate $q_C^{(2)}$. We can combine the last two terms of (12) by projecting the values of $v^{(2)}$ onto $\Gamma_C^{(1)}$ along the smoothed outward normal direction via the operator P (see e.g. Section 2.2.1 of [14] for a discussion of P) giving

$$\sum_{m=1}^{2} \left[\int_{\Omega^{(m)}} \left(k^{(m)} \nabla T^{(m)} \cdot \nabla v^{(m)} - f^{(m)} v^{(m)}\right) \mathrm{d}x + \int_{\Gamma_N^{(m)}} q_N^{(m)} v^{(m)} \, \mathrm{d}s \right] + \int_{\Gamma_C^{(1)}} q_C^{(1)} \left(v^{(1)} - P v^{(2)}\right) \mathrm{d}s = 0 \tag{13}$$

We note that (13) can be solved as written without introducing a Lagrange multiplier variable, since $q_C^{(1)}$ is given in terms of the temperature fields in the two bodies according to (6). In this case, however, we are interested in methods for

accurately computing the correct heat flux over non-matching meshes, which the mortar finite element method provides. Therefore, as is the standard practice in such methods, we introduce a Lagrange multiplier variable λ representing the normal flux at the contact interface:

$$\lambda = q_C^{(1)} \tag{14}$$

The Lagrange multiplier formulation of (13) is then: find $(T^{(1)}, T^{(2)}, \lambda) \in \mathcal{S}^{(1)} \times \mathcal{S}^{(2)} \times \mathcal{M}$ such that

$$\sum_{m=1}^{2} \left[\int_{\Omega^{(m)}} \left(k^{(m)} \nabla T^{(m)} \cdot \nabla v^{(m)} - f^{(m)} v^{(m)} \right) \mathrm{d}x + \int_{\Gamma_N^{(m)}} q_N^{(m)} v^{(m)} \, \mathrm{d}s \right] + \int_{\Gamma_C^{(1)}} \lambda \left(v^{(1)} - P v^{(2)} \right) \mathrm{d}s = 0 \tag{15}$$

$$\int_{\Gamma_C^{(1)}} \left[\lambda - h \left(T^{(1)} - P T^{(2)} \right) \right] \mu \, \mathrm{d}s = 0 \tag{16}$$

holds for all $(v^{(1)}, v^{(2)}, \mu) \in \mathcal{V}^{(1)} \times \mathcal{V}^{(2)} \times \mathcal{M}$, where $\mathcal{M}$ is the Lagrange multiplier space, which is not discussed in detail here. We note that (16) weakly imposes the condition (14) based on the constitutive relation (6).

Mortar segment-to-segment contact algorithms based on the smoothed nodal normal projection approach are known to exactly conserve linear momentum. That is, the total forces on the slave and master surfaces are in equilibrium, provided that the quadrature is carefully performed [10–13]. The angular momentum, on the other hand, is not guaranteed to be conserved. The amount by which it fails to be conserved should be "small," and methods which exactly conserve the angular momentum can be implemented.

Although Galerkin finite element formulations of the primal heat conduction equation (1) generate symmetric systems of linear algebraic equations, the formulation (15)–(16) produces a block-structured matrix whose off-diagonal blocks are non-symmetric. This is in contrast to the "classical" Lagrange multiplier method—frequently used for enforcing both Dirichlet conditions (without modifying the test space) and solution continuity—which produces symmetric off-diagonal contributions.

Also in contrast to the classical Lagrange multiplier methods, the system (Jacobian) matrix for the formulation (15)–(16) does not have an indefinite "saddle point problem" structure, and therefore mixed finite element formulations of the problem are not restricted by LBB stability concerns and stabilized finite element formulations are not required. Finally, we note that the definiteness of the matrix also ensures a larger universe of preconditioners and Krylov solvers are applicable to the system (15)–(16). In particular, the algebraic multigrid preconditioner in Hypre [17], BoomerAMG, appears to work reasonably well for this system of equations despite their asymmetry.

1.2 Nodal normal projection equations

In this section, we discuss the details of solving Eqs. (2.4.5) and (2.4.6) in Yang's dissertation [13] for the reference coordinate positions $\xi^{(2)}$ and $\xi^{(1)}$, respectively. This approach is also described in the related research article [12]. The nodal projections are central to the definition of the projection operator P used in deriving the weak formulation of the multi-body heat conduction problem (15)–(16), since they are used in the definition of the "mortar segments" along which those integrals are to be computed.

1.2.1 Projection of slave nodes onto master surface

In the notation of Yang, we have:

$$\left[N_1(\xi^{(2)}) \boldsymbol{\varphi}_1^{(2)} + N_2(\xi^{(2)}) \boldsymbol{\varphi}_2^{(2)} - \boldsymbol{\varphi}_s^{(1)} \right] \times \boldsymbol{n}_s^{(1)} = \mathbf{0} \tag{2.4.5}$$

where the superscript (1) refers to the "slave" surface, (2) refers to the "master" surface, $N_i(\xi^{(2)})$ is finite element basis function i on the master surface evaluated at $\xi^{(2)}$, $\boldsymbol{\varphi}_i^{(2)}$ is node i of the master element, $\boldsymbol{\varphi}_s^{(1)}$ is the location of the

slave node which is being projected onto the master surface, and $\boldsymbol{n}_s^{(1)}$ is the outward nodal normal vector at that point. Eq. 2.4.5 is applicable when the master element into which slave node $\boldsymbol{n}_s^{(1)}$ projects has already been found via some other search algorithm. Letting

$$\boldsymbol{x}^{(2)} \equiv N_1(\xi^{(2)})\boldsymbol{\varphi}_1^{(2)} + N_2(\xi^{(2)})\boldsymbol{\varphi}_2^{(2)} \tag{17}$$

be the physical space location on the master surface of the reference coordinate $\xi^{(2)}$, we can rewrite (2.4.5) in a potentially simpler form as:

$$\left[\boldsymbol{x}^{(2)} - \boldsymbol{\varphi}_s^{(1)}\right] \times \boldsymbol{n}_s^{(1)} = \boldsymbol{0} \tag{18}$$

Eq. (18) is in general a set of three equations in the single unknown $\xi^{(2)}$. An approximate solution to this overdetermined set of equations can be found via the least squares solution method, but we will consider the special case in which the vectors $\boldsymbol{x}^{(2)} - \boldsymbol{\varphi}_s^{(1)}$ and $\boldsymbol{n}_s$ lie in the (x_1, x_2) plane, and therefore their cross product is in the x_3-direction only. We can then obtain an explicit solution to (18), which further reduces to the scalar equation:

$$u_1 v_2 - u_2 v_1 = 0 \tag{19}$$

where

$$\boldsymbol{u} \equiv \boldsymbol{x}^{(2)} - \boldsymbol{\varphi}_s^{(1)} \tag{20}$$

$$\boldsymbol{v} \equiv \boldsymbol{n}_s^{(1)} \tag{21}$$

Substituting $N_1(\xi) \equiv \frac{1-\xi}{2}$, $N_2(\xi) \equiv \frac{1+\xi}{2}$ into (19) and rearranging yields:

$$\xi^{(2)} = \frac{-\left[\varphi_{1,1}^{(2)} + \varphi_{2,1}^{(2)} - 2\varphi_{s,1}^{(1)}\right] n_{s,2}^{(1)} + \left[\varphi_{1,2}^{(2)} + \varphi_{2,2}^{(2)} - 2\varphi_{s,2}^{(1)}\right] n_{s,1}^{(1)}}{\left[-\varphi_{1,1}^{(2)} + \varphi_{2,1}^{(2)}\right] n_{s,2}^{(1)} - \left[-\varphi_{1,2}^{(2)} + \varphi_{2,2}^{(2)}\right] n_{s,1}^{(1)}} \tag{22}$$

where subscripts after commas refer to the spatial coordinate, i.e.

$$\boldsymbol{n}_s^{(1)} \equiv \left(n_{s,1}^{(1)}, n_{s,2}^{(1)}\right) \tag{23}$$

$$\boldsymbol{\varphi}_1^{(2)} \equiv \left(\varphi_{1,1}^{(2)}, \varphi_{1,2}^{(2)}\right) \tag{24}$$

etc. A representative configuration in which a slave node is mapped onto the surface of a master element is shown in Fig. 2. This formula for computing the projection also works when the slave node $\boldsymbol{\varphi}_s^{(1)}$ lies directly on the master surface, i.e. in the case where $\boldsymbol{x}^{(2)} = \boldsymbol{\varphi}_s^{(1)}$, since this configuration satisfies (18). When the master element and the nodal normal are parallel, the denominator of (22) will be zero. This does not indicate a failure of the method, however, only that the normal projection of $\boldsymbol{\varphi}_s^{(1)}$ cannot possibly lie on the master element in question, and the implementation should therefore prevent floating point exceptions from being raised in this case.

1.2.2 Inverse-projection of master nodes from slave surface

In the converse case, we have a similar geometric situation, but need to perform an "inverse" mapping in order to determine where a given node $\boldsymbol{\varphi}_m^{(2)}$ on the master surface would have originated from, had it been projected along the slave surface nodal normal direction. The nonlinear equation to be satisfied in this case is Yang's Eq. (2.4.6) which we repeat here:

$$\left[N_1(\xi^{(1)})\boldsymbol{\varphi}_1^{(1)} + N_2(\xi^{(1)})\boldsymbol{\varphi}_2^{(1)} - \boldsymbol{\varphi}_m^{(2)}\right] \times \left[N_1(\xi^{(1)})\boldsymbol{n}_1^{(1)} + N_2(\xi^{(1)})\boldsymbol{n}_2^{(1)}\right] = \boldsymbol{0} \tag{2.4.6}$$

The unknown in (2.4.6) is $\xi^{(1)}$, and, as shown in Fig. 3, $\boldsymbol{n}_i^{(1)}$ is the nodal normal at slave surface node i,

$$\boldsymbol{x}^{(1)} \equiv N_1(\xi^{(1)})\boldsymbol{\varphi}_1^{(1)} + N_2(\xi^{(1)})\boldsymbol{\varphi}_2^{(1)} \tag{25}$$

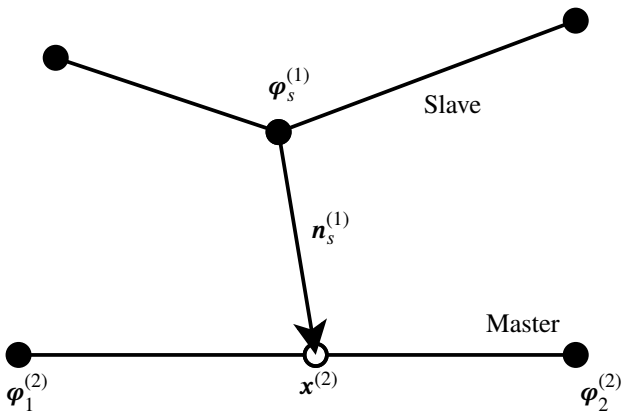


Figure 2: Diagram demonstrating projection of the slave node $\boldsymbol{\varphi}_s^{(1)}$ along the nodal normal direction onto an element on the master surface defined by the nodes $(\boldsymbol{\varphi}_1^{(2)}, \boldsymbol{\varphi}_2^{(2)})$. The solid dots represent nodes in the finite element mesh, the open dot shows where basis functions on the master surface must be evaluated during mortar segment integral computations.

is the unknown physical-space coordinate of the location where master node $\boldsymbol{\varphi}_m^{(2)}$ would have been projected from, and

$$\boldsymbol{n}(\boldsymbol{x}^{(1)}) \equiv N_1(\xi^{(1)})\boldsymbol{n}_1^{(1)} + N_2(\xi^{(1)})\boldsymbol{n}_2^{(1)} \tag{26}$$

is the outward nodal normal vector on the slave surface at that location. Using (25) and (26) allows us to write (2.4.6) more compactly as:

$$\left[\boldsymbol{x}^{(1)} - \boldsymbol{\varphi}_m^{(2)}\right] \times \boldsymbol{n}(\boldsymbol{x}^{(1)}) = \boldsymbol{0} \tag{27}$$

It is clear that relation (27) is nonlinear (quadratic) in $\boldsymbol{x}^{(1)}$ which depends on the unknown reference coordinate $\xi^{(1)}$, and therefore we do not expect to be able to solve it directly as was the case in (22).

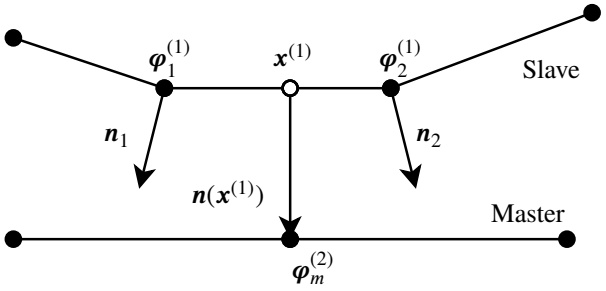


Figure 3: Diagram demonstrating "inverse" projection of the master node $\boldsymbol{\varphi}_m^{(2)}$ to the point $\boldsymbol{x}^{(1)}$ where it *would have come from* along the slave surface nodal normal direction. The solid dots represent nodes in the finite element mesh, the open dot shows the endpoint of a mortar segment used for mortar integral evaluation.

Instead, once again assuming that all vectors lie in the (x_1, x_2) plane and defining

$$\boldsymbol{u} \equiv \boldsymbol{x}^{(1)} - \boldsymbol{\varphi}_m^{(2)} \tag{28}$$

$$\boldsymbol{v} \equiv \boldsymbol{n}(\boldsymbol{x}^{(1)}) \tag{29}$$

and the residual

$$F(\xi^{(1)}) \equiv u_1 v_2 - u_2 v_1 \tag{30}$$

we will use Newton's method to compute $\xi_*^{(1)}$ such that $F(\xi_*^{(1)}) = 0$ using an initial guess of $\xi^{(1)} = 0$. Newton's method requires the first derivative of the functional, which in this case is simply

$$F'(\xi^{(1)}) \equiv (u_1 v_2' + u_1' v_2) - (u_2 v_1' + u_2' v_1) \tag{31}$$

where

$$u_i' \equiv N_1' \varphi_{1,i}^{(1)} + N_2' \varphi_{2,i}^{(1)} \tag{32}$$

$$v_i' \equiv N_1' n_{1,i} + N_2' n_{2,i} \tag{33}$$

for $i = 1, 2$, where $N_1' \equiv -\frac{1}{2}$ and $N_2' \equiv \frac{1}{2}$.

In practice, we have observed that the Newton iterations typically converge quite quickly (within 1 or 2 steps) and reliably given the initial guess of $\xi^{(1)} = 0$. When the Newton iterations converge to $|\xi^{(1)}| > 1$, this indicates that the master point in question did not come from the slave surface element we are searching, and a different candidate element should be searched instead. Finally, we observe that master nodes near the "edge" of a non-closed master surface contour may not inverse-map to *any* element on the slave surface. This does not indicate a failure of the method, however, only that the corresponding mortar segment will not have any contribution from the master side.

1.3 Discrete form of mortar integral terms

In this section, we discuss the discrete form of the mortar integral terms in (15) and (16), which for brevity we will refer to as:

$$F \equiv \int_{\Gamma_C^{(1)}} \lambda \left(v^{(1)} - P v^{(2)} \right) \, \mathrm{d}s \tag{34}$$

$$G \equiv \int_{\Gamma_C^{(1)}} \left[\lambda - h \left(T^{(1)} - P T^{(2)} \right) \right] \mu \, \mathrm{d}s \tag{35}$$

and which arise in the analysis of Section 1.1. The corresponding discrete forms of (34) and (35) are

$$F^h \equiv \int_{\Gamma_C^{(1,h)}} \lambda^h \left(v^{(1,h)} - P v^{(2,h)} \right) \, \mathrm{d}s \tag{36}$$

$$G^h \equiv \int_{\Gamma_C^{(1)}} \left[\lambda^h - h \left(T^{(1,h)} - P T^{(2,h)} \right) \right] \mu^h \, \mathrm{d}s \tag{37}$$

The discretized unknowns are written in the usual way as sums over the basis functions

$$T^{(1,h)} = \sum_{j \in \mathcal{T}^{(1)}} T_j^{(1)} N_j^{(1)} \tag{38}$$

$$T^{(2,h)} = \sum_{j \in \mathcal{T}^{(2)}} T_j^{(2)} N_j^{(2)} \tag{39}$$

$$\lambda^h = \sum_{j \in \mathcal{L}} \lambda_j M_j \tag{40}$$

where $\mathcal{T}^{(1)}$ and $\mathcal{T}^{(2)}$, are the sets of temperature (primal) variable degrees of freedom on the slave and master sides of the contact surface, respectively, and $\mathcal{L}$ is the set of the Lagrange multiplier degrees of freedom. $N_j^{(1)}$, $N_j^{(2)}$, M_j are the corresponding primal and mortar space finite element basis functions. The Lagrange multiplier is always defined on the slave (1) side, so there is no need to use a superscript for disambiguation. Correspondingly, we have

$$v^{(1,h)} = N_i^{(1)}, \quad i \in \mathcal{T}^{(1)} \tag{41}$$

$$v^{(2,h)} = N_i^{(2)}, \quad i \in \mathcal{T}^{(2)} \tag{42}$$

$$\mu^h = M_i, \quad i \in \mathcal{L} \tag{43}$$

for the test functions. Here we have assumed that the same order and family (e.g. first-order Lagrange) finite elements are used in both bodies for the primal variable, but allowed for the possibility that the Lagrange multiplier variable may be discretized with a different basis.

There will be two residual contributions (one for the master side degrees of freedom, and one for the slave side degrees of freedom) due to (36). These contributions are given by

$$F_i^{(1,h)} \equiv \int_{\Gamma_C^{(1)}} \lambda^h N_i^{(1)} \, \mathrm{d}s, \quad i \in \mathcal{T}^{(1)} \tag{44}$$

$$F_i^{(2,h)} \equiv -\int_{\Gamma_C^{(1)}} \lambda^h P N_i^{(2)} \, \mathrm{d}s, \quad i \in \mathcal{T}^{(2)} \tag{45}$$

The corresponding Jacobian contributions are

$$\frac{\partial F_i^{(1,h)}}{\partial \lambda_j} = \int_{\Gamma_C^{(1)}} M_j N_i^{(1)} \, \mathrm{d}s, \quad i \in \mathcal{T}^{(1)},\ j \in \mathcal{L} \tag{46}$$

$$\frac{\partial F_i^{(2,h)}}{\partial \lambda_j} = -\int_{\Gamma_C^{(1)}} M_j P N_i^{(2)} \, \mathrm{d}s, \quad i \in \mathcal{T}^{(2)},\ j \in \mathcal{L} \tag{47}$$

Likewise, the residual contribution of (37) due to Lagrange multiplier test function i is given by

$$G_i^h \equiv \int_{\Gamma_C^{(1)}} \left[\lambda^h - h\left(T^{(1,h)} - PT^{(2,h)}\right)\right] M_i \, \mathrm{d}s, \quad i \in \mathcal{L} \tag{48}$$

with corresponding Jacobian contributions

$$\frac{\partial G_i^h}{\partial T_j^{(1)}} = -\int_{\Gamma_C^{(1)}} h M_i N_j^{(1)} \, \mathrm{d}s, \quad i \in \mathcal{L},\ j \in \mathcal{T}^{(1)} \tag{49}$$

$$\frac{\partial G_i^h}{\partial T_j^{(2)}} = \int_{\Gamma_C^{(1)}} h M_i P N_j^{(2)} \, \mathrm{d}s, \quad i \in \mathcal{L},\ j \in \mathcal{T}^{(2)} \tag{50}$$

$$\frac{\partial G_i^h}{\partial \lambda_j} = \int_{\Gamma_C^{(1)}} M_i M_j \, \mathrm{d}s, \quad i, j \in \mathcal{L} \tag{51}$$

Terms (44), (46), (49), and (51) depend only on values from the slave side, and are straightforward to compute using the existing discretization of the slave boundary. On the other hand, the terms (45), (47), (48), and (50) depend on values from *both* the slave and master sides, and quadrature errors due to the integration of locally piecewise continuous functions will result if we attempt to compute the integral using only the slave side discretization.

1.4 Computing the mortar integral: Potential inaccuracies

To understand the potential accuracy issues involved with computing the mortar integral contributions, we consider a simple example in which a mesh with equal-sized elements is "perfectly unaligned" with a mesh of equal-sized elements on the other side of the contact interface as in Fig. 4. For simplicity, we also assume the mortar and primal variables are discretized with the same finite element basis, so that $M_i = N_i^{(1)}\ \forall i$, and that the heat transfer coefficient is simply $h = 1$ in the following discussion.

In this case, the nodal normal projection operation is trivial, and the master side basis functions are defined piecewise on slave side element Ω_e as:

$$N_a^{(2)} = \begin{cases} 1 + \xi^{(1)}/2, & -1 \le \xi^{(1)} \le 0 \\ 1 - \xi^{(1)}/2, & 0 \le \xi^{(1)} \le 1 \end{cases} \tag{52}$$

$$N_b^{(2)} = \begin{cases} 0 & -1 \le \xi^{(1)} \le 0 \\ \xi^{(1)}/2, & 0 \le \xi^{(1)} \le 1 \end{cases} \tag{53}$$

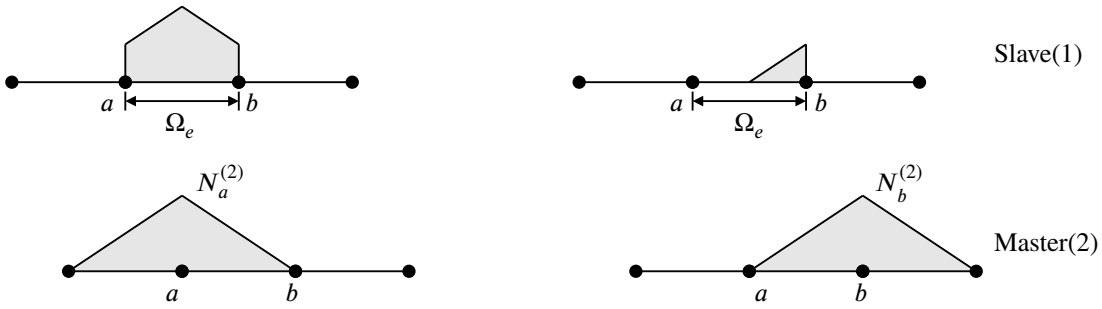


Figure 4: Master side basis functions $N_a^{(2)}$ and $N_b^{(2)}$ on slave side element Ω_e in the perfectly unaligned mesh case. Two of the three master side basis functions with support on Ω_e are shown, the last one is symmetric to $N_b^{(2)}$.

while the slave side basis functions are defined in the usual way as

$$N_a^{(1)} = \frac{1-\xi^{(1)}}{2} \tag{54}$$

$$N_b^{(1)} = \frac{1+\xi^{(1)}}{2} \tag{55}$$

The master/slave contributions due to e.g. (47) (ignoring the sign) can be organized in matrix form as:

$$K_{ms} = \begin{bmatrix} \int_{\Omega_e} N_a^{(2)} N_a^{(1)} \mathrm{d}s & \int_{\Omega_e} N_a^{(2)} N_b^{(1)} \mathrm{d}s \\ \int_{\Omega_e} N_b^{(2)} N_a^{(1)} \mathrm{d}s & \int_{\Omega_e} N_b^{(2)} N_b^{(1)} \mathrm{d}s \end{bmatrix} \tag{56}$$

The true values (using exact integration) of K_{ms} are given by

$$K_{ms} = \frac{h_e}{2} \begin{bmatrix} \frac{3}{4} & \frac{3}{4} \\ \frac{1}{24} & \frac{5}{24} \end{bmatrix} \tag{57}$$

where h_e is the length of element Ω_e. Using standard two-point quadrature, on the other hand, results in

$$\widetilde{K}_{ms} \approx \frac{h_e}{2} \begin{bmatrix} 0.711325 & 0.711325 \\ 0.061004 & 0.227671 \end{bmatrix} \tag{58}$$

This corresponds to a relative error

$$\frac{\|K_{ms} - \widetilde{K}_{ms}\|_\infty}{\|K_{ms}\|_\infty} \approx 5\% \tag{59}$$

Although relatively small in magnitude, there is no way to know, for a given problem, how strongly this error will affect the solution, and it will cause the usual global conservation properties of the finite element method to be lost. For instance, in contact mechanics problems, linear momentum will not be conserved if the mortar integrals are computed in this way; i.e. spurious momentum will be introduced in the gap between the two subdomains. Likewise, for thermal contact problems, the total energy will not be conserved.

1.5 Computing the mortar integral: Simple examples

We next consider in detail the computation of mortar segment contributions for the simple, perfectly unaligned mesh in Fig. 5. In particular, we will compute the slave/slave and master/slave element stiffness contributions for segment S in Fig. 5. In this example, we assume the slave and master reference coordinates are aligned in opposite directions

so that $\xi_a^{(1)} = 0$, $\xi_b^{(1)} = 1$, $\xi_a^{(2)} = 1$, $\xi_b^{(2)} = 0$. As discussed in Yang's dissertation [13], we parameterize the segment S with the secondary reference variable $-1 \leq \eta \leq 1$ such that

$$\xi^{(1)}(\eta) = \xi_a^{(1)} \frac{1-\eta}{2} + \xi_b^{(1)} \frac{1+\eta}{2} \tag{60}$$

$$\xi^{(2)}(\eta) = \xi_a^{(2)} \frac{1-\eta}{2} + \xi_b^{(2)} \frac{1+\eta}{2} \tag{61}$$

and again assume that the mortar and primal variables are discretized with the same finite element basis, which allows us to write the slave/slave and master/slave Jacobian contributions due to this segment as

$$K_{ss}(i,j) \equiv \int_S N_i^{(1)} N_j^{(1)} \mathrm{d}s = \frac{h_{\text{seg}}}{2} \int_{-1}^{1} N_i^{(1)}(\xi^{(1)}(\eta)) N_j^{(1)}(\xi^{(1)}(\eta)) \, \mathrm{d}\eta \tag{62}$$

$$K_{ms}(i,j) \equiv \int_S N_i^{(2)} N_j^{(1)} \mathrm{d}s = \frac{h_{\text{seg}}}{2} \int_{-1}^{1} N_i^{(2)}(\xi^{(2)}(\eta)) N_j^{(1)}(\xi^{(1)}(\eta)) \, \mathrm{d}\eta \tag{63}$$

where h_{seg} is the length of segment S, and the local indices $i, j = 1, 2$ correspond to the basis functions of the (single) slave and master elements which contribute to this segment (see Fig. 5). The basis functions are defined exactly as in (54) and (55), with indices a and b corresponding to 1 and 2. The method of constructing the mortar segments guarantees that a unique element from each side contributes values to each segment, and that the integrands (62) and (63) are continuous (and therefore exactly integrable using, in this case, a two-point quadrature rule).

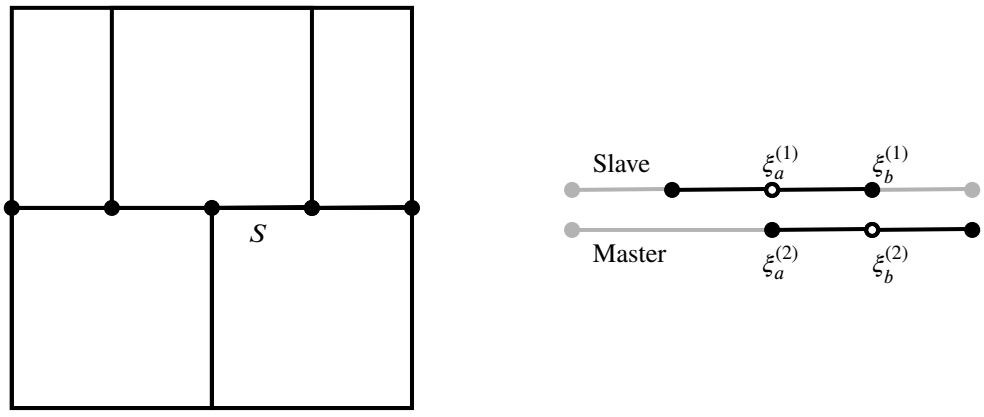


Figure 5: Perfectly unaligned mesh with five elements used for verification (left) and mortar segment detail for the segment marked S (right) with endpoints $\xi_a^{(1)} = 0$, $\xi_b^{(1)} = 1$, $\xi_a^{(2)} = 1$, $\xi_b^{(2)} = 0$. Dark lines indicate the faces of the elements whose degrees of freedom S contributes to, light grey lines indicate neighboring element faces which are not involved in the segment computation.

In this simple example, the slave/slave and master/slave Jacobian contributions can be computed explicitly as follows: the segment reference coordinate values are given by (60) and (61) as

$$\xi^{(1)}(\eta) = \frac{1+\eta}{2} \tag{64}$$

$$\xi^{(2)}(\eta) = \frac{1-\eta}{2} \tag{65}$$

and consequently the slave and master side basis functions in segment reference coordinates are

$$N_1^{(1)} = \frac{1-\xi^{(1)}(\eta)}{2} = \frac{1-\eta}{4} \tag{66}$$

$$N_2^{(1)} = \frac{1+\xi^{(1)}(\eta)}{2} = \frac{3+\eta}{4} \tag{67}$$

$$N_1^{(2)} = \frac{1-\xi^{(2)}(\eta)}{2} = \frac{1+\eta}{4} \tag{68}$$

$$N_2^{(2)} = \frac{1+\xi^{(2)}(\eta)}{2} = \frac{3-\eta}{4} \tag{69}$$

This leads to the following slave/slave

$$K_{ss}(1,1) = \frac{h_{\text{seg}}}{2}\int_{-1}^{1}\frac{(1-\eta)^2}{16}\,\mathrm{d}\eta = \frac{h_{\text{seg}}}{12} \tag{70}$$

$$K_{ss}(1,2) = K_{ss}(2,1) = \frac{h_{\text{seg}}}{2}\int_{-1}^{1}\frac{(1-\eta)(3+\eta)}{16}\,\mathrm{d}\eta = \frac{h_{\text{seg}}}{6} \tag{71}$$

$$K_{ss}(2,2) = \frac{h_{\text{seg}}}{2}\int_{-1}^{1}\frac{(3+\eta)^2}{16}\,\mathrm{d}\eta = h_{\text{seg}}\frac{7}{12} \tag{72}$$

and master/slave

$$K_{ms}(1,1) = \frac{h_{\text{seg}}}{2}\int_{-1}^{1}\frac{(1+\eta)(1-\eta)}{16}\,\mathrm{d}\eta = \frac{h_{\text{seg}}}{24} \tag{73}$$

$$K_{ms}(1,2) = K_{ms}(2,1) = \frac{h_{\text{seg}}}{2}\int_{-1}^{1}\frac{(1+\eta)(3+\eta)}{16}\,\mathrm{d}\eta = h_{\text{seg}}\frac{5}{24} \tag{74}$$

$$K_{ms}(2,2) = \frac{h_{\text{seg}}}{2}\int_{-1}^{1}\frac{(3-\eta)(3+\eta)}{16}\,\mathrm{d}\eta = h_{\text{seg}}\frac{13}{24} \tag{75}$$

Jacobian contributions. We note that the master/slave Jacobian contribution is not symmetric in general, it simply happens to be in this particular case. In summary, we have

$$K_{ss} = \frac{h_{\text{seg}}}{24}\begin{bmatrix}2 & 4\\ 4 & 14\end{bmatrix} \qquad K_{ms} = \frac{h_{\text{seg}}}{24}\begin{bmatrix}1 & 5\\ 5 & 13\end{bmatrix} \tag{76}$$

and, although the values above were obtained using exact integration, it is easy to verify that the same result can be computed using two-point Gaussian quadrature since the integrands are continuous.

1.6 Test cases

In the following subsections, we verify the correctness and accuracy of the proposed multibody heat conduction finite element formulation using the following test problem:

$$-\nabla^2 u + u = f \quad \in \Omega \tag{77}$$

$$\frac{\partial u}{\partial n} = g \quad \in \partial\Omega \tag{78}$$

where $\Omega = \left(-\frac{1}{2}, \frac{1}{2}\right)^2$ is the unit square centered at the origin, and the forcing $f = -4 + x^2 + y^2$ is chosen to give the true solution $u = x^2 + y^2$ in the single-body case. The Neumann boundary data $g = -1$ is chosen to be consistent with the true solution on the entire boundary.

In the following examples, we cut the domain described above in various ways, and introduce a small gap to implement the multi-body heat transfer problem. Therefore, the true solution will not actually be $u = x^2 + y^2$ in these examples,

and we do not compute *a posteriori* estimates of the error. The heat transfer coefficient is taken to be

$$h = \frac{k}{\ell} \tag{79}$$

where $k = .03$ is a constant, and ℓ is the width of the gap between the bodies.

In the following sections, we will refer to discretizations which use a piecewise continuous linear approximation for the primal variable combined with a piecewise discontinuous constant approximation for the Lagrange multiplier variable as $P^1 - P^0$ discretizations, and to discretizations that use a continuous linear Lagrange multiplier approximation as $P^1 - P^1$ discretizations. In the present work, we do not consider quadratic (P^2) approximations for the primal variable, however these discretizations are of interest in future work.

1.6.1 Flat interface

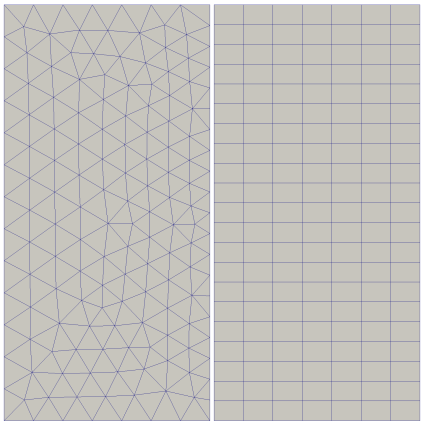
In this test case, a gap of width .01 is introduced along the plane $x = 0$ in Ω, as shown in Fig. 6a. The left side of the domain is meshed with three-noded triangular elements while the right side is mesh with four-noded quadrilaterals. The left side of the domain is (arbitrarily chosen to be the slave side. The solution computed on this split domain is shown in Fig. 6b. The contour lines of the solution are approximately aligned along the gap in this case since the flux of the true solution is nearly zero in the gap. The flux solutions on successively refined grids are shown in Fig. 6c for the $P^1 - P^0$ discretization, and 6d for the $P^1 - P^1$ discretization.

We expect the true flux to converge to a constant value in this case, which is proportional to the size of the gap. Both the $P^1 - P^0$ and $P^1 - P^1$ discretizations of the flux exhibit small numerical oscillations, both in the interior of the mesh, and near the boundary, but the scale of these oscillations is quite small, and they are damped out as the mesh is refined. Both discretizations appear to be of comparable accuracy as the mesh is refined, but the $P^1 - P^1$ discretization has more error than the $P^1 - P^0$ discretization on the coarsest mesh. Both methods required a similar number of GMRES iterations to solve, and this number grew only modestly under mesh refinement while using a standard ILU(1) preconditioner.

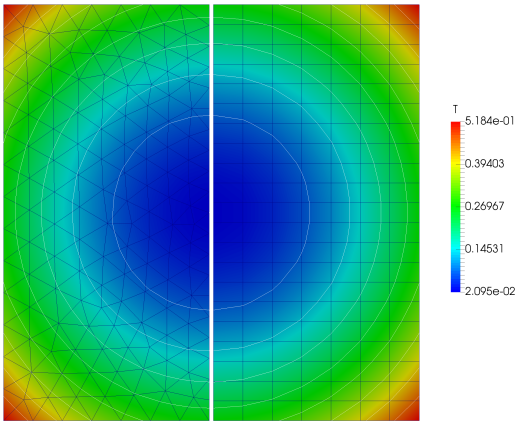
1.6.2 Quarter-circle interface

In this test case, we introduce a quarter-circular gap in the domain Ω corresponding to concentric circles centered at $(-1/2, -1/2)$ with radii $R = 0.7$ and $R = \sqrt{2}/2$. The coarse level mesh for this case, which has approximately 20 elements along the curved interface, is shown in Fig. 7a. We choose the $R = 0.7$ surface as the slave side, and mesh the inside of the circular region with three-noded triangular elements and the outside with four-noded quadrilaterals. The solution computed on this domain is shown in Fig. 7b, and in this case there is a nonzero flux along the entire quarter-circular arc, so we do *not* expect the temperature fields on either side of the gap to be continuous as in the flat interface case.

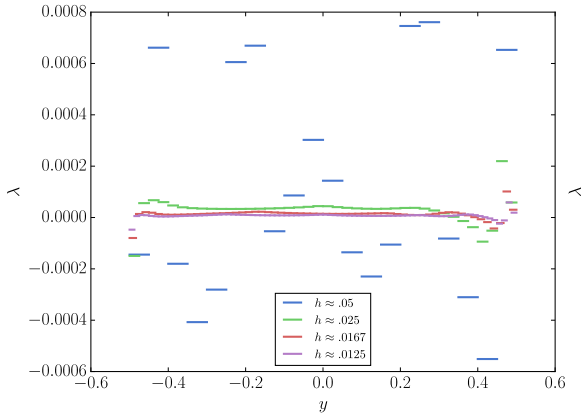
The computed fluxes are plotted for a sequence of refined grids in Fig. 7c for the $P^1 - P^0$ discretization, and 7d for the $P^1 - P^1$ discretization. It appears that both discretizations converge to same true solution, and, in contrast to the previous case, we do not observe any obvious numerical oscillations in this case, although that may simply be due to the fact that the flux is overall much larger in this test. The biggest error in the flux occurs near the $x = -0.5$ limit of the domain for both discretizations, but this error quickly converges under mesh refinement. As in the previous case, the linear systems associated to both the $P^1 - P^0$ and $P^1 - P^1$ discretizations were solved in a modest number of GMRES iterations using both the ILU(1) and BoomerAMG preconditioners, as well as the direct `-pc_type lu` solver which is distributed with PETSc.



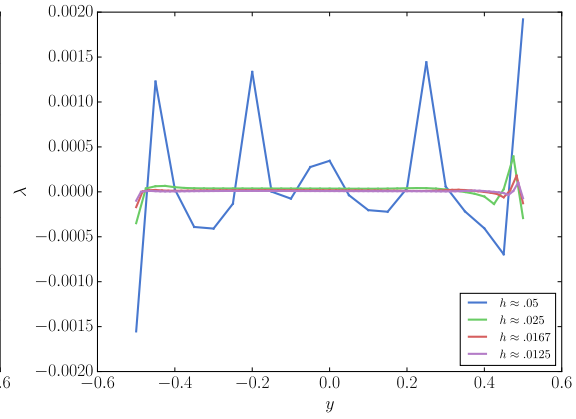
(a) Hybrid mesh with element size $h \approx 1/20$.



(b) Temperature field.

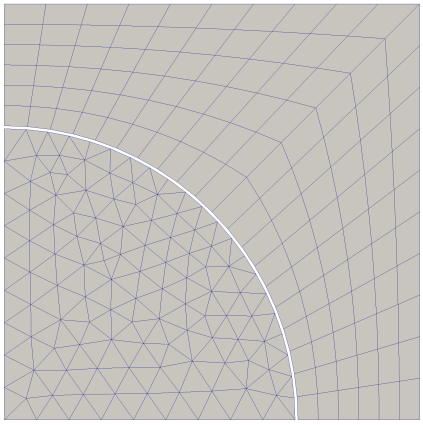


(c) λ, $P^1 - P^0$ discretization.

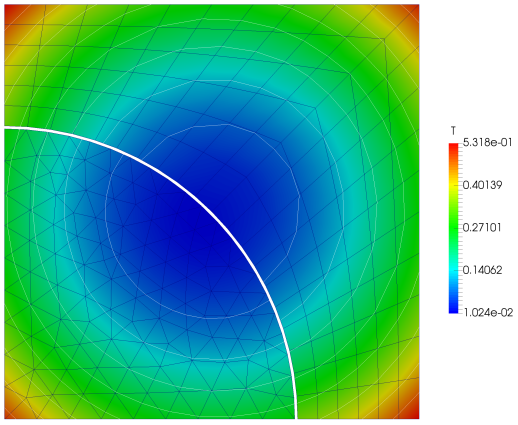


(d) λ, $P^1 - P^1$ discretization.

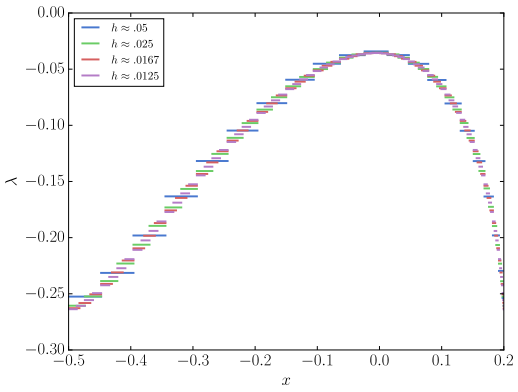
Figure 6: Hybrid mesh (a), temperature solution field (b), and flux solutions for the $P^1 - P^0$ (c) and $P^1 - P^1$ (d) discretizations of the flat interface test case.



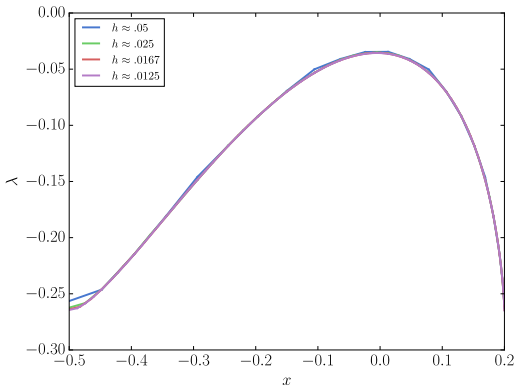
(a) Hybrid mesh with element size $h \approx 1/20$.



(b) Temperature field.



(c) λ, $P^1 - P^0$ discretization.



(d) λ, $P^1 - P^1$ discretization.

Figure 7: Hybrid mesh (a), temperature solution field (b), and flux solutions for the $P^1 - P^0$ (c) and $P^1 - P^1$ (d) discretizations of the quarter-circle interface test case.

2 Augmented Lagrange Multiplier Method for Mechanical Contact Enforcement

2.1 Introduction

In the current version of BISON, penalty and kinematic formulations are implemented to enforce node-to-surface contact constraints. In the case of penalty enforcement, the normal contact force is computed as a product of a penalty parameter and the penetration distance. The tangential contact force is solved to satisfy the penalized slip rule. The accuracy of the contact enforcement improves as the penalty parameter is increased. However, with very large penalty parameters, the linear system suffers from ill-conditioning and becomes difficult to solve. The use of a penalty formulation usually results in some amount of penetration and makes the simulation results less accurate. On the other hand, the kinematic formulation transfers the residual from the slave node to the master surface and penalizes the iterative gap such that penetration is removed in the converged solution. The kinematic formation works robustly in BISON for both frictionless and glued contact. In FY16, a hybrid formulation was developed to further improve the robustness of frictional contact. The hybrid formulation uses a kinematic approach to strictly enforce the non-penetration condition and a penalty method in the tangential direction to enforce the slipping constraints. Although the hybrid formulation gives more robust solutions, it still results in some errors because of the penalty regularization in the tangential direction which permits a small amount of relative sliding before slip begins.

The contact module in MOOSE has been significantly improved over the past few years, but it still suffers several numerical issues. As an alternative formulation, the Augmented Lagrangian Method (ALM) has enjoyed considerable success in the treatment of contact problems and it is desirable to implement this formulation in MOOSE as an additional alternative to the present methods. The ALM approach combines either the penalty method or the constitutive law with Lagrange multiplier methods and results in robust solutions and smaller errors.

2.2 Algorithmic Developments

The ALM algorithm implemented in MOOSE is demonstrated in Figure 8. In the first iteration in a given time step, Γ_{stick} is taken to be all of Γ and further amendments to Γ_{stick} are made in the augmentation phase, according to the current state of the multipliers. The Coulomb frictional contact condition is enforced in the augmentation phase which makes the nonlinear solve converge much better. The nested ALM is implemented as a customized `FEProblem`, called `AugmentedLagrangianContactProblem`, which checks for the constraint satisfaction and decides if convergence is achieved or to repeat the step. The related parameters called `penetration_tolerance`, `tangential_increment_stick_tolerance` and `tangential_friction_force_tolerance` correspond to TOL1, TOL2 and TOL3 in Figure 8, respectively.

2.3 Numerical Examples

2.3.1 Sliding Sphere (3D)

Hertz contact between a sphere and a rigid surface is a classic problem within contact mechanics. A 3D model was setup in BISON and the geometry and mesh is shown in Figure 9. The rigid block is fixed in all directions and normal and tangential displacement boundary conditions are applied to the surface nodes of the half sphere.

This problem was previously run under frictional contact conditions with the penalty method and tangential penalty (i.e., hybrid) formulation with certain limitations. To converge, the penalty parameters have to be relatively small and result in a significant amount of penetration. In addition, the incremental loading is chosen to be small and the sphere can only slip over a small tangential distance. In contrast, the newly implemented ALM formulation can run with five times larger loading increments and displace the sphere almost to the end of the rigid block without any convergence issues.

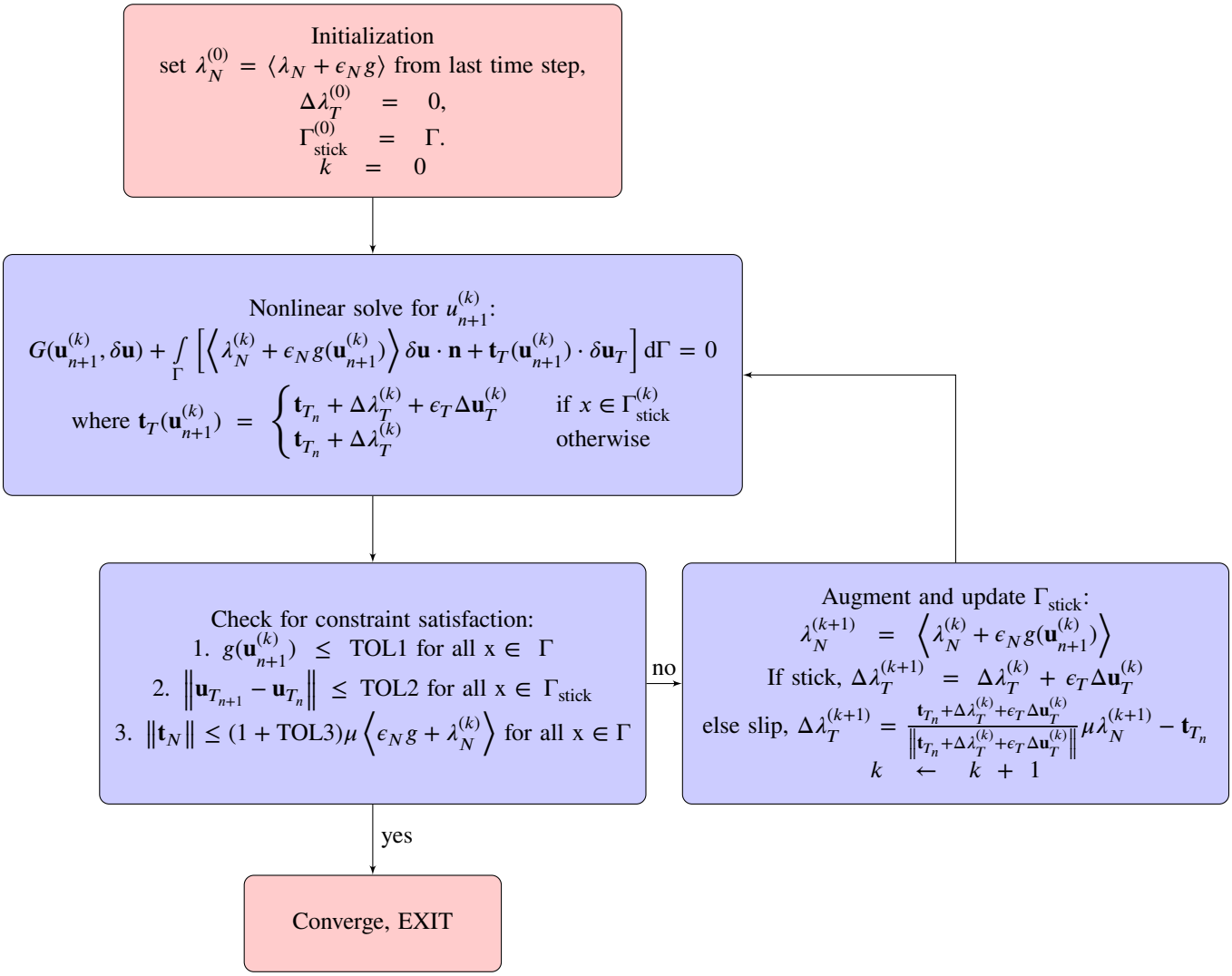


Figure 8: Nested augmented Lagrangian algorithm for frictional contact in MOOSE.

The simulation was run with a frictional coefficient of 1.0. A displacement boundary condition is applied at the top surface of the sphere. The contour of σ_{zz} shown in Figure 10 is fairly smooth. The quantity of reaction tangential force divided by the normal force is plotted in Figure 11. In the first stage, the sphere is push down and does not have any tangential force. In the second stage, the reaction tangential force is less than the frictional capacity and results in a sticking condition. In the last stage, the sphere starts to slip and the tangential force stays the same as the normal force as predicted by the coulomb friction law.

Calvert Cliffs

The Calvert Cliffs assessment test cases are a useful set of full length 2D RZ models for testing newly implemented features in BISON. The UFE019 test case used in this instance is a full length rod ($\sim$ 3.8 m) irradiated for four cycles in reactor and then discharged with a burnup of $\sim$ 47 GWd/MTU. A range of contact conditions have been used to model the fuel/cladding contact for this case are the effects on the cladding elongation is shown in Figure 12.

As expected the frictionless case underpredicts the cladding elongation due to the lack of frictional interaction between the fuel and cladding. The glued contact case predicts the highest elongation during operation since the fuel and

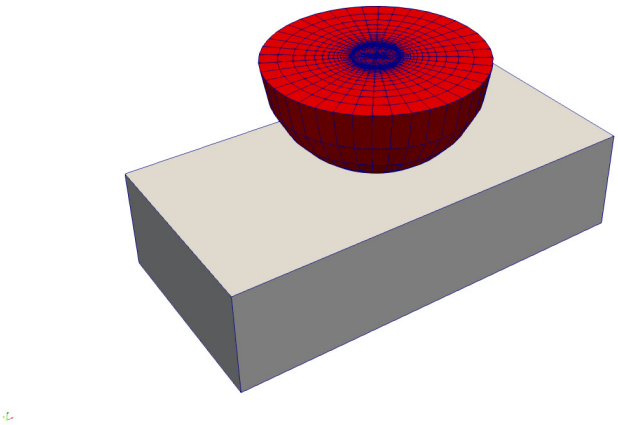


Figure 9: Geometry and mesh of 3D sphere sliding on block problem.

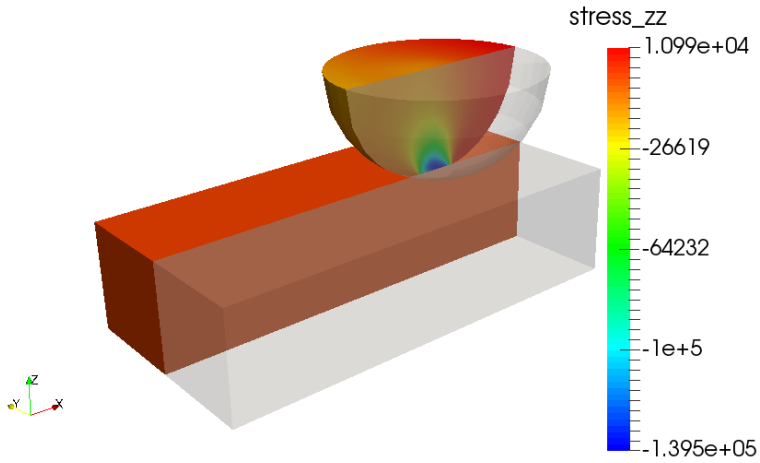


Figure 10: Stress zz contour of indenter.

cladding are "bonded" once they come into contact. However, during the cool down (i.e., power down) the fuel contracts and pulls the cladding as well to again underpredict the measured elongation. The two contact models using Coulomb friction ($\mu = 0.4$) with different implementations produce the most realistic behavior. Both the tangential penalty and augmented Lagrange methods match the glued contact behavior during the in-reactor phase of the simulation, but allow some relative sliding of the fuel and cladding during the cool down. Therefore, the final cladding elongation predictions for those models is much closer to the measured value.

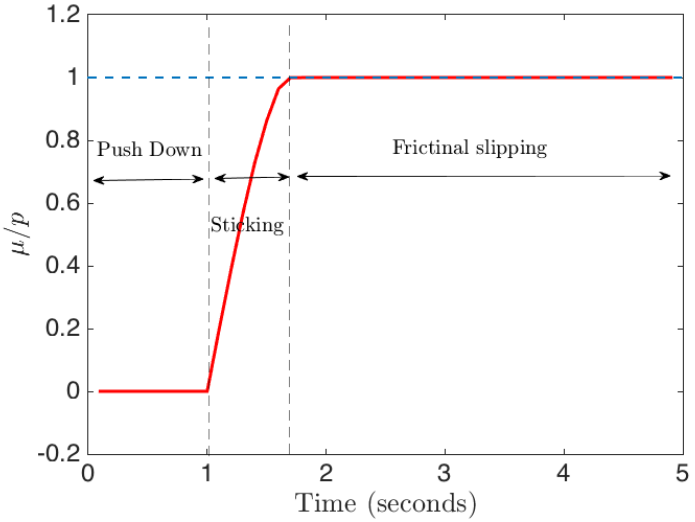


Figure 11: The tangential and normal reaction force with frictional coefficient 1.0

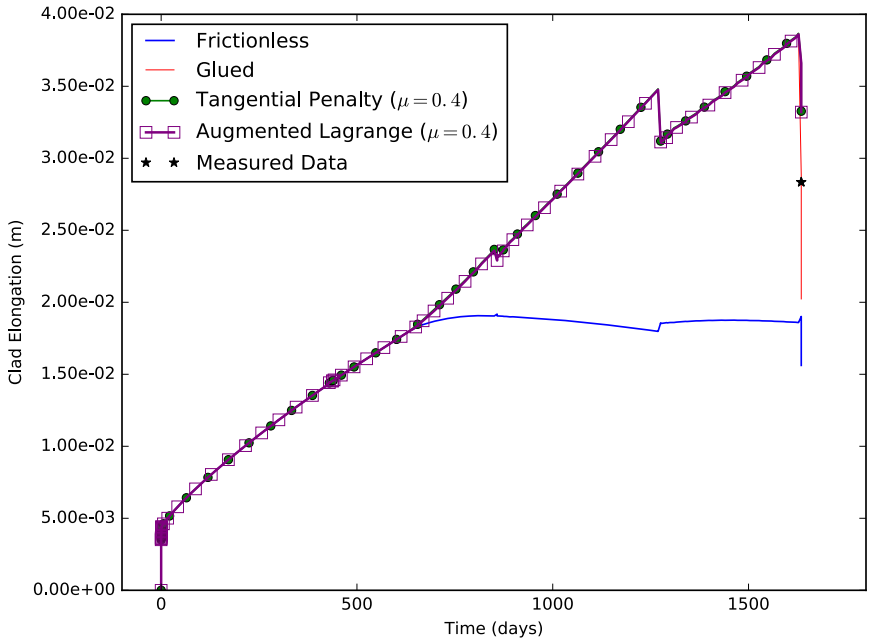


Figure 12: Clad elongation history for Calvert Cliffs test case UFE019 showing comparison of contact models.

3 Improvements to Contact Search Robustness

Node-to-surface contact is the widely used contact type in BISON for simulating contact between a fuel rod and the cladding. In these cases, the outer surface of the fuel pellets form the slave boundary, which is discretized into a set of slave nodes, and the inner surface of the cladding forms the master boundary, which is discretized into a set of element faces. The aim of contact search algorithms is to efficiently and robustly determine the master element face at which a given slave node is likely to come in contact.

In the current approach, a contact patch, which is a subset of the master boundary, is initially saved for each slave node. This patch contains the k (patch size) nodes in the master boundary that are closest to the slave node. In the subsequent time steps or iterations, the one master node that is closest to the slave node is obtained from among the nodes in this saved contact patch. All the master element faces that are connected to this closest master node are then checked to see if the slave node is in contact with any of these master element faces. If no contact is detected among these selected master element faces, the slave node is determined as not in contact with the master boundary.

The method of saving off a contact patch for each slave node and running subsequent contact searches on only this saved patch reduces the computational time as the whole master boundary does not have to be searched at every time step/iteration. But as the slave/master boundaries move during the simulation, the contact patch has to be updated accordingly to accurately detect contact. The users are currently given two options of updating the patch — auto and always. In the 'always' option, the contact patch for all slave nodes is updated at the start of every time step. This can be computationally expensive. In the 'auto' option, the contact patch is updated only when the closest master node for any of the slave nodes fall close to the edge of the contact patch. This check to determine if the closest master node is near the edge of the contact patch (and patch update if required) is performed at the start of every time step. In certain fuel problems, there can be substantial slipping between the fuel pellets and cladding from one nonlinear iteration to the next within a given time step. In these situations, updating the patch only at the start of the time step may lead to erroneous contact detection, because slave nodes may have slid out of the patch within a step. This can happen even if the patch is updated every step if there is large sliding within a step. This is more likely to occur with more refined meshes, becuase the same relative mesh movement occurs over a larger number of faces.

To improve contact search efficiency and robustness, the following modifications have been made to the contact search algorithm:

3.1 Use of k-d trees for nearest neighbor search

The computational time required to create the contact patch increases with an increase in the number of nodes in the slave and master boundary, and with an increase in the contact patch size. The brute force method to find the k nearest neighbors to a given slave node from the master boundary is to first find the distance between the slave node and each master node. These distances then have to be sorted in increasing order and the first k nodes then form the contact patch for that slave node. The nearest neighbor search that was used in MOOSE/BISON before the work described here employed a priority queue data structure which optimizes the distance sorting time and works faster than the brute force method. Both these methods are more suitable when fewer slave nodes are present. K-d trees are a much faster search method for problems with larger number of nodes in the slave and master boundary.

K-d trees work by dividing the master nodes into bins/leaves at different depths. For example, if the master boundary is a rectangle as shown in Fig. 13, it is first divided into two bins (left and right at depth 1). These bins can then be subdivided into two more bins (depth 2) and this subdivision continues until the maximum number of nodes in each bin/leaf of the tree is less than a user specified number. Once this tree is created, the nearest neighbor search for each slave node is executed by traversing down along the tree starting at depth 1. The Nanoflann k-d tree library [18] has been used to improve the contact patch creation in MOOSE. This approach improves contact search time by a factor of 1.38 and 1.45, respectively, for the case where 1000 and 2000 nodes, respectively, are considered in both the master and slave boundaries. When 2000 nodes are present in the slave boundary and 30,000 nodes are present in the master boundary, the k-d tree approach is 90 times faster than the priority queue approach.

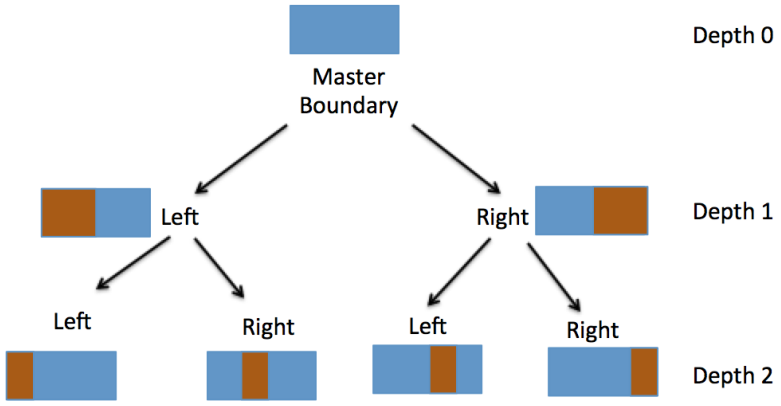


Figure 13: Sample k-d tree for a simplistic rectangular master boundary. The brown region is the region represented by given bin/leaf.

3.2 Updating contact patch during nonlinear iterations

As mentioned earlier, the contact patch could only be updated at the start of the time step using the 'always' or 'auto' patch update strategy. To avoid cases where the slave node slips outside the contact patch during nonlinear iterations within a given time step, a new patch update strategy has been added that checks and updates the patch in every nonlinear iteration for a subset of the slave nodes. All the slave nodes which are not in contact with the master boundary are placed in a set of 'recheck_slave_nodes'. The nearest neighbor search is called to update the patch for all the nodes contained in 'recheck_slave_nodes'. The closest master node and all the element faces connected to closest master node are then obtained for each of these slave nodes and contact detection is re-run on these new master faces. Some of these nodes in the 'recheck_slave_nodes' set may not actually be in contact with a face, but this ensures that contact is correctly enforced on any nodes that are in contact with a face. When used with the k-d tree, this approch ensures that contact is accurately detected during all stages of the simulation with a minimal increase in computational effort.

These improvements in the contact search were tested on a 2-D fuel problem with 10 fuel pellets (Figure 14) with frictionless contact between the fuel and the clad. The slave (outer surface of fuel pellets) and master boundary (inner surface of clad) both contain 320 nodes. Different contact options were tested with changes in patch size and patch update strategies. The solution from the run with 'always' patch update strategy with patch size of 10 is assumed to be the correct solution against which the solution from the other cases are compared (because the patch is sufficiently large to accomodate the sliding in a step). The solution quantities compared at each time step are average temperature of the fuel rod and cladding, cladding and pellet volume, fission gas produced and released, gas volume, flux from clad and fuel, total and input rod power.

Table 1 gives the time taken to run the fuel-clad contact problem using 10 processors with 3 different patch update strategies (always, auto and the newly added nonlinear iteration patch update strategy) and with 3 different patch sizes (1, 5 and 10). For all the cases except the 'auto' and 'always' patch update strategy with patch size of 1, the results match well with the correct solution with relative differences below 1e-5. There are small differences in the time taken in running the different cases. The timings show that for a given patch size, the 'nonlinear updte' strategy is computationally less expensive than the other two strategies. This is probably because in this strategy, the patch is updated only for a small subset of the slave nodes – those that have moved outside the patch – as opposed to the other two strategies where the patch is regularly updated for all the 320 slave nodes. The nonlinear patch update strategy also works well with arbitrary patch sizes while the other two methods do not work with a patch size of 1 because nodes easily slide out of that patch during the solution of a time step.

Using a large patch is detrimental to performance on large problems because storage is pre-allocated for off-diagonal Jacobian entries related to the contact constraints for all elements in the patch. For much larger contact problems, using the nonlinear iteration patch update strategy with a small patch size will likely give greatly improved performance

Figure 14: 2D problem with 10 discrete fuel pellets. During the course of the simulation, the fuel pellets (red) expand and come in contact with the cladding (blue).

Table 1: Timing comparison of contact patch update strategies on fuel-clad contact problem

Patch update strategy	**Patch size = 1**	**Patch size = 5**	**Patch size = 10**
always	Failed	627.11 s	625.56 s
auto	Failed	606.2 s	614.1 s
nonlinear iteration	596.97s	603.75 s	611.45 s

relative to the current strategy of using a sufficiently large patch to prevent sliding outside the patch.

4 Summary and Future Work

This report documents improvements to BISON's contact enforcement algorithms in three areas: mortar enforcement of thermal contact, augmented Lagrange enforcement of mechanical contact, and contact search improvements. The mortar method for thermal contact described here has currently only been demonstrated on fairly simple demonstration problems, but is near to a point in development where it can be applied to thermal contact enforcement in 2D fuel simulations. The results shown here demonstrate that this method has good rates of convergence, without oscillations in the solution. Future development will focus on applying this for thermal contact enforcement in LWR fuel performance models, and enabling the use of this technique on 3D models. In addition, this method can also be applied to improve the accuracy and robustness of mechanical contact enforcement.

The Augmented Lagrange Multiplier approach has been implemented and applied to two problems with frictional contact, a 3D Hertz contact model and a 2D full-length fuel rod. This algorithm gave robust convergence on these models, both of which have been challenging with existing algorithms in BISON. This code is ready to be tested to assess its performance on a wider range of BISON PCMI problems of interest, both 2D and 3D.

The improvements to the contact search address longstanding issues where contact could potentially not be enforced on nodes that slide too far during a time step, which can occur during a PCMI simulation. This can also potentially improve performance by permitting the use of a smaller patch size, which decreases the number of pre-allocated off-diagonal Jacobian entries. Work is underway to allow for those off-diagonal entries to be dynamically added or removed from the solver within a time step as the set of contact constraints evolves.

The combination of the work described here in three independent areas is expected to greatly improve the accuracy and robustness of the algorithms used to enforce both thermal and mechanical contact in BISON. These developments are in varying levels of maturity. Mortar enforcement is an entirely new capability, and the algorithms described here were developed from the ground-up, and are not yet committed to the code base. Initial results look very promising, but this is a capability that still needs more work before it is ready for full production. The augmented Lagrange method builds on the existing node/face enforcement, and is ready for expanded testing. The core components of the search improvements have been committed to the code base. The next steps in development for those search improvments is to perform expanded testing on larger models, and change the default code behavior based on findings from that testing.

References

[1] C. Hesch and P. Betsch. Energy-momentum consistent algorithms for dynamic thermomechanical problems—Application to mortar domain decomposition methods. *International Journal for Numerical Methods in Engineering*, 86(11):1277–1302, June 2011. http://tinyurl.com/rm646r4.

[2] I. Babuška. The finite element method with Lagrange multipliers. *Numerische Mathematik*, 20(3):179–192, June 1973. https://doi.org/10.1007/BF01436561.

[3] H. J. C. Barbosa and T. J. R. Hughes. The finite element method with Lagrange multipliers on the boundary: Circumventing the Babuška-Brezzi condition. *Computer Methods in Applied Mechanics and Engineering*, 85(1):109–128, January 1991. https://doi.org/10.1016/0045-7825(91)90125-P.

[4] F. Ben Belgacem. The mortar finite element method with Lagrange multipliers. *Numerische Mathematik*, 84(2):173–197, December 1999. http://tinyurl.com/rdrfu2e.

[5] M. G. Larson and A. Massing. L^2-error estimates for finite element approximations of boundary fluxes. ArXiv e-print, September 2014. http://arxiv.org/abs/1401.6994.

[6] J. C. Simo, P. Wriggers, and R. L. Taylor. A perturbed Lagrangian formulation for the finite element solution of contact problems. *Computer Methods in Applied Mechanics and Engineering*, 50(2):163–180, August 1985. http://dx.doi.org/10.1016/0045-7825(85)90088-X.

[7] P. Papadopoulos and R. L. Taylor. A mixed formulation for the finite element solution of contact problems. *Computer Methods in Applied Mechanics and Engineering*, 94(3):373–389, February 1992. http://dx.doi.org/10.1016/0045-7825(92)90061-N.

[8] T. A. Laursen and J. C. Simo. A continuum-based finite element formulation for the implicit solution of multibody, large deformation frictional contact problems. *International Journal for Numerical Methods in Engineering*, 36(20):3451–3485, October 1993. http://dx.doi.org/10.1002/nme.1620362005.

[9] T. A. Laursen. *Computational Contact and Impact Mechanics*. Springer-Verlag, 2002.

[10] M. A. Puso. A 3D mortar method for solid mechanics. *International Journal for Numerical Methods in Engineering*, 59(3):315–336, January 2004. http://dx.doi.org/10.1002/nme.865.

[11] M. A. Puso and T. A. Laursen. A mortar segment-to-segment contact method for large deformation solid mechanics. *Computer Methods in Applied Mechanics and Engineering*, 193(6–8):601–629, February 2004. http://dx.doi.org/10.1016/j.cma.2003.10.010.

[12] B. Yang, T. A. Laursen, and X. Meng. Two dimensional mortar contact methods for large deformation frictional sliding. *International Journal for Numerical Methods in Engineering*, 62(9):1183–1225, March 2005. http://tinyurl.com/yba8fa36.

[13] B. Yang. *Mortar finite element methods for large deformation contact mechanics*. PhD thesis, Duke University, 2006. http://tinyurl.com/gkt5nga.

[14] M. Gitterle. *A dual mortar formulation for finite deformation frictional contact problems including wear and thermal coupling*. PhD thesis, Technische Universität München, November 2012. http://tinyurl.com/rrv57o7.

[15] S. Hüeber and B. I. Wohlmuth. Thermo-mechanical contact problems on non-matching meshes. *Computer Methods in Applied Mechanics and Engineering*, 198(15–16):1338–1350, March 2009. http://tinyurl.com/tsn3lh7.

[16] S. Falletta and B. P. Lamichhane. Mortar finite elements for a heat transfer problem on sliding meshes. *Calcolo*, 46(2):131–148, June 2009. http://tinyurl.com/sgr37xf.

[17] R. D. Falgout and U. M. Yang. hypre: A library of high performance preconditioners. In P. M. A. Sloot, A. G. Hoekstra, C. J. K. Tan, and J. J. Dongarra, editors, *Proceedings of the International Conference on Computational Science (ICCS 2002)*, pages 632–641, Amsterdam, The Netherlands, April 21–24, 2002. Springer Berlin Heidelberg. https://doi.org/10.1007/3-540-47789-6_66.

[18] Jose Luis Blanco and Pranjal Kumar Rai. nanoflann: a C++ header-only fork of FLANN, a library for nearest neighbor (NN) wih kd-trees. `https://github.com/jlblancoc/nanoflann`, 2014.