A Novel Face-on-Face Contact Method for Nonlinear Solid Mechanics

By

STEVEN ROBERT WOPSCHALL

B.S. (University of California, San Diego) 2006

M.S. (University of California, San Diego) 2010

DISSERTATION

Submitted in partial satisfaction of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

Civil and Environmental Engineering

in the

OFFICE OF GRADUATE STUDIES

of the

UNIVERSITY OF CALIFORNIA

DAVIS

Approved:

Mark Rashid, Chair

N. Sukumar

John Bolander

Committee in Charge

2017

Abstract

The implicit solution to contact problems in nonlinear solid mechanics poses many difficulties. Traditional node-to-segment methods may suffer from locking and experience contact force chatter in the presence of sliding. More recent developments include mortar based methods, which resolve local contact interactions over face-pairs and feature a kinematic constraint in integral form that smoothes contact behavior, especially in the presence of sliding. These methods have been shown to perform well in the presence of geometric nonlinearities and are demonstratively more robust than node-to-segment methods. These methods are typically biased, however, interpolating contact tractions and gap equations on a designated non-mortar face, which leads to an asymmetry in the formulation. Another challenge is constraint enforcement. The general selection of the active set of constraints is frought with difficulty, often leading to non-physical solutions and easily resulting in missed face-pair interactions. Details on reliable constraint enforcement methods are lacking in the greater contact literature.

This work presents an unbiased contact formulation utilizing a median-plane methodology. Up to linear polynomials are used for the discrete pressure representation and integral gap constraints are enforced using a novel subcycling procedure. This procedure reliably determines the active set of contact constraints leading to physical and kinematically admissable solutions void of hueristics and user action.

The contact method presented herein successfully solves difficult quasi-static contact problems in the implicit computational setting. These problems feature finite deformations, material nonlinearity, and complex interface geometries, all of which are challenging characteristics for contact implementations and constraint enforcement algorithms. The subcycling procedure is a key feature of this method, handling active constraint selection for complex interfaces and mesh geometries.

Acknowledgments

My wife once asked me if I look at the gallery wall she put up in the home office. While I told her that I continually enjoy looking at the dozen or so pictures of mountains and landscapes, I was thinking of a piece of advice received early on in my first meeting with my advisor: do not be afraid to stare at the wall and think, possibly for hours. While it started with mere words in a simple statement, Mark Rashid has demonstrated, and cultivated my own, dedication to thought. Standing on the other side of this journey, I cannot help but rephrase the statement: do not be afraid, this work takes time. His time, as it turns out, was something Mark worked very hard to give his students. Weekly research meetings, lessons in code development, a new course on computational mechanics, a new research code, lunches to discuss mechanics and politics, and meetings at national labs are but few of the tangible manifestations of his willingness to make students part of *his* work. Mark has shown me that an idea or thought is not born out of correctness or incorrectness, but rather exists in a state that reflects one's own understanding and knowledge. As one's understanding and knowledge develop and evolve, the idea itself develops and evolves and is given meaning. Mark has demonstrated the effect of his time and dedication to mechanics through his precision of thought and deep understanding of theory, mathematics, and computation. For these reasons I am grateful for his time, patience, and guidance.

I want to thank Joe Jung, formerly of Sandia National Laboratories, for the opportunity to join his solid mechanics code development group as a summer intern in 2013, and for his support of the research conducted by Mark Rashid and his students. In the midst of a massive code base and computational complexities, Joe was quick to share and excite enthusiasm saying, "this stuff is really cool." I also learned that there is nothing better to encite comraderie and ensure positive group morale than an office espresso machine.

Additional thanks is owed to Rose McCallen of Lawrence Livermore National Laboratory.

While on sabbatical at UC Davis, Rose was instrumental in providing networking introductions and career advice, both of which played a role in my choice to pursue a postgraduate career at LLNL.

My graduate experience was additionally defined by working along side Mark Rashid's students, Omar Hafez and Brian Giffin. First and foremost, thanks must be given to Omar for the Mooney-Rivlin hyperelastic constitutive model and to Brian for his postprocessing work, both of which were implemented in our group's research code and utilized in this work. Additional thanks is given to Omar for his work in helping create and run the Computational Mechanics Working Group. We started CMWG in order to bring together students in computational mechanics for the purposes of networking, socializing, exchanging ideas, presenting research, and gaining exposure to the work of faculty members, industry leaders, and researchers at national labs. A final thanks is extended to Omar for the many hours spent together at the white board working through theory and problems in preparation for our Qualifying exams.

Lastly, this was a journey within a journey and my deepest thanks is extended to my wife, Allison. She has helped me to be stronger, more disciplined, and more balanced throughout experiences that easily provoke the opposite. Her own strength and understanding has been one of my greatest sources of comfort and motivation. While she was quick to interpret minimizing functionals and coding on my computer as "minimizing fun" and writing in "pretty colors," she has always acknowledged that my engineering work is a part of who I am and who I want to become. In thanking my wife, I must also thank our daughter, Charlotte. There is no truer a marker of time than welcoming your first born into the world and watching her grow. There is no precise thought that does justice to what she means to me.

Contents

1 Overview

Computational contact mechanics is an integral part of engineering solid mechanics simulations; however, contact has historically been a difficult numerical problem. This is especially true for implicit, quasi-static analysis. For this reason, robust numerical methods modeling contact behavior have significant impact on the engineering industry. The contact problem itself can be divided into three areas: fast and robust contact search and detection; contact discretization, pressure representation and constraint formulation; and contact constraint enforcement. This work presents a novel face-on-face contact methodology that specifically addresses the last two areas.

A contact formulation includes the precise spatial discretization methodology used to form the contact interface. The discretization not only defines the regions over which contact integrals are evaluated and unknown contact pressures are specified, but also provides the geometric basis for the constraint formulation. Many difficulties exist depending on the selection of a discretization method and constraint formulation, as well as the pressure representation and constraint enforcement method. Difficulties include non-smooth contact solutions and overconstraint, numerical instabilities, and unphysical (interpenetration or interface tension) solutions, respectively. While face-on-face methods have been shown to successfully address contact smoothness and overconstraint, there is much variation to be had in the precise formulation of these methods, thus precipitating further study. Additionally, limitations on the pressure representation have been shown to adequately address numerical instabilities, yet advantages associated with a higher order pressure representation are not well understood. Lastly, a clear, reliable constraint enforcement method leading to physically correct, kinematically admissible contact solutions, independent of hueristics and user action, likely represents one of the greater numerical challenges associated with a computational contact implementation. This difficulty is further compounded by the implicit solution setting in

which this work is performed.

This work presents a median-plane methodology for constructing discrete, "fictitious" contact surfaces from local face-pair interactions. Up to linear polynomial pressure representations are used on each local surface to enforce the zero-interpenetration constraint between contacting face-pairs. This *normal* kinematic constraint is expressed in integral form enforcing zero-mean gap between two opposing faces across their discrete contact surface. While this work primarily addresses frictionless contact, an extension of the forumlation and enforcement technique is provided to account for Coulomb friction.

A primary difficulty still exists in determining the exact distribution of face-pairs over which to enforce zero-interpenetration. This work does so by introducing a subcycling procedure that cycles over the discrete constraints in a way that produces physically correct, kinematically admissible contact solutions, free of hueristics or user action. The contact methodology presented herein was designed and developed for implicit, nonlinear quasi-static analysis and was implemented in an implicit finite element research code.

The contact methodology presented in this work allows for the solution of difficult implicit, quasi-static contact problems. These problems feature material and geometric nonlinearity, large sliding, and complex interface geometries. These characteristics make the solution of implicit, quasi-static contact problems challenging for all contact methods and impossible for some. Moreover, the constraint enforcement procedure provides a clear, reliable and problem independent method for determining the active set of contact constraints without user action. These features result in a robust contact implementation for the solution of nonlinear quasi-static contact problems.

Prior to a proper introduction to numerical methods in contact and an associated literature review, a number of theoretical concepts must first be covered in the sections that follow. Section 2 covers notation and mathematical preliminaries pertinent to this work. Then,

Section 3 introduces the nonlinear theory of solid mechanics prior to the presentation of the finite element approximation to the nonlinear solid mechanics problem, which is presented in Section 4. Following this, Section 5 covers nonlinear solution strategies. At this point, enough theoretical groundwork has been laid to introduce the large deformation contact problem, which is presented in Section 6. This sets the stage for the discussion and review of numerical methods in contact found in Section 7, which contextualizes the problem and establishes the contact vernacular prior to the presentation of the novel method proposed in this work. Section 8 presents the contact discretization used in this work, while the mathematical formulation of the proposed method follows in Section 9. Section 10 discusses the numerical implementation and subcycle procedure, while numerical examples demonstrating the efficacy of this method are presented in Section 11. Future work and concluding remarks are presented in Sections 12 and 13.

2 Notation and Mathematical Preliminaries

In this work, we will consider frictionless contact mechanics in a finite deformation setting, including material nonlinearity. We will begin with some notation and mathematical preliminaries that lead into an introduction to nonlinear solid mechanics.

2.1 Notation

2.1.1 Vectors and Tensors

In this work, vectors and tensors will be boldface, and scalars will appear in regular typeset. We will largely be dealing with finite deformations, and will therefore be interested in the reference and current configurations of a body. To distinguish between the two, a position vector in the reference configuration will be uppercase, and a position vector in the current configuration will be lower case, e.g. $\mathbf{X}$ and $\mathbf{x}$, respectively. Tensors will always be uppercase, e.g. $\mathbf{A}$, and the context will distinguish between a tensor and a vector in the reference configuration.

2.1.2 Index Notation and Einstein Summation

Component indices in the reference configuration will be lowercase Greek letters, each ranging from 1 to 3. For example, X_α, $\alpha = 1, 2, 3$ represents the three components of the reference configuration vector, $\mathbf{X}$. Alternatively, vector indices in the current configuration will be lowercase Latin letters each ranging from 1 to 3. For example, x_i, i = 1, 2, 3, represents the three components of the current configuration vector, $\mathbf{x}$. Each Greek and Latin letter is a dummy index. Therefore α, β, γ, etc., and i, j, k, etc., are interchangeable and will be used

throughout the text. Tensors will have two indices, either Greek lowercase letters or Latin lowercase letters, or one of both, depending on whether that specific index is referring to the reference or current configuration. For example, $S_{\alpha\beta}$, $P_{i\alpha}$, and T_{ij}, are written where all of the components of $\mathbf{S}$ are in the reference configuration, the first index of $\mathbf{P}$ is in the current configuration, while the second index is in the reference configuration, and both indices of $\mathbf{T}$ are in the current configuration.

Einstein summation notation will be used with our index notation. In general, summation will be over repeated indices. For example, a vector dot product in index notation is

$$a_i v_i = c. \tag{2.1}$$

Note the summation is over the repeated index, i, and the scalar result, c, on the right hand side has no indices associated with it, as it is a scalar. A matrix-vector product in index notation is

$$A_{ij} v_j = x_i. \tag{2.2}$$

Note the summation is over the repeated index j, resulting in vector components with index i. Note that any special notation or operators that come out of Einstein notation will be explained in the text.

2.2 Mathematical Preliminaries

Let us consider a scalar field, $\phi(\mathbf{x})$, vector field, $\mathbf{v}(\mathbf{x})$, and tensor field, $\mathbf{A}(\mathbf{x})$, $\forall\, \mathbf{x} \in \mathbb{R}^n, n = 1, 2, \text{or } 3$, that are assumed to be continuous and differentiable on domain Ω.

2.2.1 Basis Vectors

Considering a Cartesian coordinate system in $\mathbb{R}^3$ with coordinates $\{x_1, x_2, x_3\}$, we define the associated basis vectors $\mathbf{e}_i = \{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3\}$ where

$$\mathbf{e}_i = \frac{\partial \mathbf{x}}{\partial x_i}. \tag{2.3}$$

Since the Cartesian unit basis vectors are aligned with the coordinate directions, orthonormality holds. That is,

$$\mathbf{e}_i \cdot \mathbf{e}_j = \delta_{ij}, \tag{2.4}$$

where δ_{ij} is the Kronecker delta, which is defined as

$$\begin{aligned} \delta_{ij} =& 1, \text{if i} = \text{j} \\ & 0, \text{if i} \neq \text{j}. \end{aligned} \tag{2.5}$$

We may write our vector, $\mathbf{v}(\mathbf{x})$, and tensor, $\mathbf{A}(\mathbf{x})$, in terms of components as

$$\begin{aligned} \mathbf{v} &= v_i \mathbf{e}_i \\ \mathbf{A} &= A_{ij} \mathbf{e}_i \otimes \mathbf{e}_j, \end{aligned} \tag{2.6}$$

where $\otimes$ is the outer, or dyadic, product.

2.2.2 Comma Notation for Partial Derivatives and Differentiation in Time

Partial differentiation in space will be denoted by the following subscript notation,

$$\frac{\partial u_i}{\partial x_i} = u_{i,i}, \frac{\partial u_i}{\partial x_j} = u_{i,j}, \ \frac{\partial A_{ij}}{\partial x_j} = A_{ij,j}, \frac{\partial \phi}{\partial x_j} = \phi_{,j}. \tag{2.7}$$

Time derivatives will have a dot above the variable. For example

$$\frac{\partial \mathbf{v}}{\partial t} = \dot{\mathbf{v}}, \tag{2.8}$$

2.2.3 Vector and Tensor Operations

The following is a table of vector and tensor operations taken from [1].

Operation	**Direct Notation**	**Index Notation**
vector dot product	$\mathbf{v} \cdot \mathbf{a}$	$v_i a_i$
vector cross product	$\mathbf{v} \times \mathbf{a}$	$\epsilon_{ijk} v_i a_j$
dyadic product	$\mathbf{v} \otimes \mathbf{a}$	$v_i a_j$
tensor inner product	$\mathbf{Av}$	$A_{ij} v_j$
”	$\mathbf{vA}$	$A_{ji} v_j$
”	$\mathbf{e}_i \otimes \mathbf{e}_j \mathbf{e}_k$	$\delta_{jk} \mathbf{e}_i$
”	$\mathbf{AB}$	$A_{ij} B_{jk}$
”	$\mathbf{BA}$	$B_{ij} A_{jk}$
transpose	$\mathbf{A} = \mathbf{B}^T$	$A_{ij} = B_{ji}$
symmetry	$\mathbf{A} = \mathbf{A}^T$	$A_{ij} = A_{ji}$
antisymmetry	$\mathbf{A} = -\mathbf{A}^T$	$A_{ij} = -A_{ji}$
trace	$tr\mathbf{A}$	A_{ii}
determinant	$\det\mathbf{A}$	$\frac{1}{6} \epsilon_{ijk} \epsilon_{lmn} A_{il} A_{jm} A_{kn}$

where the alternator symbol, ϵ_{ijk}, is defined

$$\begin{aligned} \epsilon_{ijk} = &+1, \text{if } ijk \text{ is an even permutation of 1,2,3;} \\ &-1, \text{if } ijk \text{ is an odd permutation of 1,2,3;} \\ &0, \text{if } ijk \text{ is not a permutation of 1,2,3.} \end{aligned} \tag{2.9}$$

2.2.4 Divergence, Gradient, and Curl

Let the gradient of a scalar valued function be defined as

$$\text{grad}\ \phi(\mathbf{x}) = \nabla\phi = \frac{\partial\phi(\mathbf{x})}{\partial x_p}\mathbf{e}_p. \tag{2.10}$$

Let the gradient of a vector valued function be defined as

$$\text{grad}\ \mathbf{u}(\mathbf{x}) = \nabla\mathbf{u}(\mathbf{x}) = \frac{\partial\ u_p}{\partial x_q}\mathbf{e}_p \otimes \mathbf{e}_q. \tag{2.11}$$

The divergence of a vector valued function is defined as

$$\text{div}\ \mathbf{u}(\mathbf{x}) = \nabla \cdot \mathbf{u}(\mathbf{x}) = \frac{\partial u_p}{\partial x_p}. \tag{2.12}$$

The curl of a vector valued function is defined as

$$\text{curl}\ \mathbf{u}(\mathbf{x}) = \epsilon_{pqr}\frac{\partial u_r}{\partial x_q}\mathbf{e}_p. \tag{2.13}$$

The divergence of a second rank tensor is defined as

$$\text{div}\ \mathbf{A}(\mathbf{x}) = \nabla \cdot \mathbf{A}(\mathbf{x}) = \frac{\partial A_{pq}}{\partial x_p}\mathbf{e}_{\ q}. \tag{2.14}$$

2.2.5 Divergence Theorem

The divergence theorem relates the volume integral of the divergence of a vector field to the surface integral of the normal component of that vector field. The theorem states that for a vector field $\mathbf{v}(\mathbf{x})$,

$$\int_\Omega \nabla \cdot \mathbf{v} dv = \int_{\partial\Omega} \mathbf{v} \cdot \mathbf{n} da, \tag{2.15}$$

which in index notation is

$$\int_{\Omega} v_{i,i} dv = \int_{\partial\Omega} v_i n_i da, \tag{2.16}$$

where $\mathbf{n}$ is the vector normal to the boundary of the body, $\partial\Omega$. Note that the vector field must be at least piecewise differentiable on Ω, whereas the normal vector, $\mathbf{n}$, must be at least piecewise continuous on $\partial\Omega$.

2.2.6 Polar Decomposition of a Second Rank Tensor

It is useful to define the decomposition of an invertible second order tensor, $\mathbf{F}$ as follows: For all $\mathbf{F}$ such that $\det(\mathbf{F}) > 0$, there exists a $\mathbf{U}$ and $\mathbf{R}$ such that

$$\mathbf{F} = \mathbf{R} \cdot \mathbf{U} = \mathbf{V} \cdot \mathbf{U}, \tag{2.17}$$

where $\mathbf{R}^{-1} = \mathbf{R}^T$ and $\det(\mathbf{R}) = 1$. Furthermore, $\mathbf{U} = \mathbf{U}^T$, $\mathbf{V} = \mathbf{V}^T$, and $\mathbf{x}^T\mathbf{U}\mathbf{x}$, $\mathbf{x}^T\mathbf{V}\mathbf{x} > 0 \, \forall \, \mathbf{x} \in \mathbb{R}^n, \mathbf{x} \neq \mathbf{0}$, i.e. $\mathbf{U}$ and $\mathbf{V}$ are symmetric positive definite.

2.2.7 Localization Theorem

Let $\mathbf{R}(\mathbf{x})$ be a continuous and real, vector-valued, function defined on the open domain, $\Omega \in \mathbb{R}^3$. Let ω be an arbitrary subdomain contained in Ω. That is, $\omega \subset \Omega$. The localization theorem states that

$$\int_{\omega} \mathbf{R}(\mathbf{x}) dv = \mathbf{0} \quad \forall \ \omega \subset \Omega \Rightarrow \mathbf{R}(\mathbf{x}) = \mathbf{0} \ \forall \quad \mathbf{x} \in \Omega. \tag{2.18}$$

This theorem will be an important component in our derivation of the equations of motion in nonlinear solid mechanics.

3 Introduction to Nonlinear Solid Mechanics

In this work, we are considering the contact mechanics problem in the context of finite deformations. As a result, we need to develop the basics for the nonlinear continuum description of the deformation of a solid, including appropriate kinematic quantities and stress measures.

3.1 Problem Preliminaries: Nonlinear Kinematics of a Continuum

Let us consider an arbitrary solid body that occupies an open region denoted by Ω, with closure, $\bar{\Omega}$. The body undergoes a deformation from an initial or reference configuration to a current configuration. Let Ω_0 denote the open region of the body in the reference configuration, and $\bar{\Omega}_0$ its closure. Let Ω_n denote the open region of the body in the current configuration at an arbitrary time, $t_n \in [0, \mathrm{T}], \mathrm{T} > 0$, and $\bar{\Omega}_n$ its closure. Let $\mathbf{X} \in \bar{\Omega}_0$ be a vector denoting the position of a material point in the body's initial, or reference configuration. The components of $\mathbf{X}$ are X_α, where Greek letters, e.g. $\alpha, \beta, \gamma = 1, 2, 3$ are dummy indices referring to the reference configuration. Additionally, let $\mathbf{x} \in \bar{\Omega}_n$ be a vector denoting the position of the same material point in the current configuration. The components of this vector are x_i, where Latin letters, e.g. i, j, k = 1, 2, 3 are dummy indices referring to the current configuration. Furthermore, let $\mathbf{x} = \mathbf{x}(\mathbf{X},\mathrm{t})$, which may alternatively be expressed as the mapping $\varphi_n : \mathbf{X} \to \mathbf{x}(\mathbf{X}, t_n)$. See Figure 1 for a representation of the reference configuration and the current configuration, as described.

As can be seen in Figure 1, the vector describing the displacement of a material point in the body between the reference configuration and the current configuration is

$$\mathbf{u(X)} = \mathbf{x}(\mathbf{X}, t) - \mathbf{X}. \tag{3.1}$$

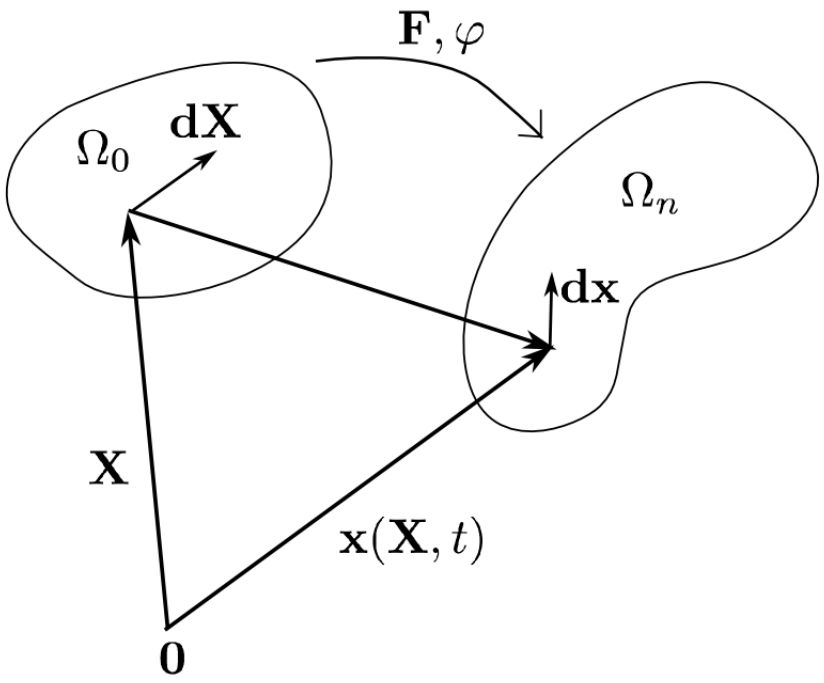

Figure 1: Representation of an arbitrary body in the reference and current configurations.

Additionally, Figure 1 shows an infinitesimal material line segment in the reference and current configurations, denoted d$\mathbf{X}$ and d$\mathbf{x}$, respectively, which will come into play in the next section. Furthermore, the figure shows that the local motion may be described by the deformation gradient $\mathbf{F}$, which maps material line segments from the reference to the current configuration, or φ, which maps material points from the reference to current configuration, i.e. $\varphi : \mathbf{X} \to \mathbf{x}$. The deformation gradient specifically will be important in the discussion of nonlinear kinematics, and will be discussed in more detail in the next section.

3.2 Finite Strain Kinematics

3.2.1 Deformation Gradient

Let us define the gradient of $\mathbf{x}(\mathbf{X},\mathrm{t})$ as:

$$\mathbf{F} = \frac{\partial \mathbf{x}}{\partial \mathbf{X}}. \tag{3.2}$$

Furthermore, given a set of basis vectors in the current configuration, $\mathbf{e}_i = \{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3\}$, and a set of basis vectors in the reference configuration, $\mathbf{E}_\alpha = \{\mathbf{E}_1, \mathbf{E}_2, \mathbf{E}_3\}$, the deforamtion gradient can be written in component for as

$$\begin{aligned} &\mathbf{F} = \frac{\partial x_i}{\partial X_\alpha} \mathbf{e}_i \otimes \mathbf{E}_\alpha \\ &\text{or,} \\ &F_{i\alpha} = \frac{\partial x_i}{\partial X_\alpha}. \end{aligned} \tag{3.3}$$

$\mathbf{F}$ maps the inifinitesimal line segment in the reference configuration, $\mathrm{d}\mathbf{X}$, to the infinitesimal line segment in the current configuration, $\mathrm{d}\mathbf{x}$, in the following way,

$$\mathrm{d}\mathbf{x} = \mathbf{F}\mathrm{d}\mathbf{X}. \tag{3.4}$$

Since $\mathbf{F}$ is invertible,

$$\mathrm{d}\mathbf{X} = \mathbf{F}^{-1}\mathrm{d}\mathbf{x}. \tag{3.5}$$

Then, the displacement gradient with respect to the reference configuration is

$$\frac{\partial \mathbf{u}}{\partial \mathbf{X}} = \frac{\partial \mathbf{x}}{\partial \mathbf{X}} - \mathbf{I} = \mathbf{F} - \mathbf{I}, \tag{3.6}$$

and the displacement gradient with respect to the current configuration is

$$\frac{\partial \mathbf{u}}{\partial \mathbf{x}} = \mathbf{I} - \frac{\partial \mathbf{X}}{\partial \mathbf{x}} = \mathbf{I} - \ \mathbf{F}^{-1}. \tag{3.7}$$

Lastly, the local volume change of a solid due to a particular deformation is a function of the deformation gradient and is defined as follows,

$$\frac{\mathrm{dV}}{\mathrm{dV}_0} = \det(\mathbf{F}) = J. \tag{3.8}$$

3.2.2 Strain Tensor

Consider material length measures,

$$\mathrm{dS} = \text{length of } \mathrm{d}\mathbf{X},$$
$$\mathrm{ds} = \text{length of } \mathrm{d}\mathbf{x}.$$

Then,

$$\mathrm{dS}^2 = \mathrm{d}\mathbf{X} \cdot \mathrm{d}\mathbf{X} = \mathrm{d}\mathbf{X} \cdot \mathbf{I} \cdot \mathrm{d}\mathbf{X},$$
$$\mathrm{ds}^2 = \mathrm{d}\mathbf{x} \cdot \mathrm{d}\mathbf{x} = (\mathbf{F}\mathrm{d}\mathbf{X}) \cdot (\mathbf{F}\mathrm{d}\mathbf{X}) = \mathrm{d}\mathbf{X} \cdot \ \mathbf{F}^{\mathrm{T}}\mathbf{F} \cdot \mathrm{d}\mathbf{X}$$

Let $\mathbf{C} = \mathbf{F}^{\mathrm{T}}\mathbf{F}$ be the *right Cauchy-Green* deformation tensor, then,

$$\mathrm{ds}^2 = \mathrm{d}\mathbf{X} \cdot \mathbf{C}\mathrm{d}\mathbf{X}.$$

Therefore, if we combine the last two results for the squares of the material line segments dS and ds, we have

$$\mathrm{ds}^2 - \mathrm{dS}^2 = \mathrm{d}\mathbf{X} \cdot (\mathbf{C} - \mathbf{I})\mathrm{d}\mathbf{X}, \tag{3.9}$$

where we obtain the definition of the *Green strain* (or *Lagrangian Strain*) tensor,

$$\mathbf{E} = \frac{1}{2}(\mathbf{C} - \mathbf{I}). \tag{3.10}$$

As a result, Equation 3.9 becomes

$$\mathrm{ds}^2 - \mathrm{dS}^2 = 2\mathrm{d}\mathbf{X} \cdot \mathbf{E}\mathrm{d}\mathbf{X}. \tag{3.11}$$

Note that the Green strain tensor is symmetric, because both $\mathbf{F}^{\mathrm{T}}\mathbf{F}$ and $\mathbf{I}$ are symmetric. Lastly, it is useful to state the relation,

$$\mathbf{C} = \mathbf{F}^{\mathrm{T}}\mathbf{F} = \mathbf{U}^2, \tag{3.12}$$

where $\mathbf{U}$ is from the polar decomposition, $\mathbf{F} = \mathbf{R}\mathbf{U}$.

3.2.3 Velocity Gradient

Define the spatial velocity as follows:

$$\dot{\mathbf{x}} = \frac{\partial \mathbf{x}}{\partial t} = \mathbf{v}. \tag{3.13}$$

Furthermore, define the spatial velocity gradient as

$$\mathbf{L} = \mathrm{grad}\, \dot{\mathbf{x}} = \mathrm{grad}\, \mathbf{v} = \frac{\partial \mathbf{v}}{\partial \mathbf{x}}. \tag{3.14}$$

In component form, the velocity gradient is written

$$L_{ij} = \frac{\partial v_i}{\partial x_j}. \tag{3.15}$$

To relate $\mathbf{L}$ to the deformation gradient, $\mathbf{F}$, consider the following. Let us define

$$\text{grad}\,\mathbf{u} = \frac{\partial \mathbf{u}}{\partial \mathbf{X}}.$$

Then, we can take the spatial gradient of the material to obtain

$$\text{grad}\,\dot{\mathbf{x}} = \text{grad}\,\mathbf{v} = \frac{\partial \mathbf{v}}{\partial \mathbf{X}} = \frac{\partial \mathbf{v}}{\mathbf{x}} \cdot \frac{\partial \mathbf{x}}{\partial \mathbf{X}} = \text{grad}\,\mathbf{v} \cdot \mathbf{F},$$

where,

$$\text{grad}\,\mathbf{v} = \mathbf{L} = \frac{\partial \mathbf{v}}{\partial \mathbf{x}} = \frac{\partial \mathbf{v}}{\partial \mathbf{X}} \cdot \frac{\partial \mathbf{X}}{\partial \mathbf{x}} = \text{grad}\,\mathbf{v} \cdot \mathbf{F}^{-1}.$$

Combining results and noting that grad $\mathbf{v} = \frac{\partial \dot{\mathbf{x}}}{\partial \mathbf{X}} = \dot{\mathbf{F}}$ leads to the following relation for the velocity gradient in terms of the deformation gradient:

$$\mathbf{L} = \dot{\mathbf{F}}\mathbf{F}^{-1}. \tag{3.16}$$

When we decompose the velocity gradient tensor into its symmetric and antisymmetric parts, we have

$$\mathbf{L} = \tfrac{1}{2}(\mathbf{L} + \mathbf{L}^T) + \tfrac{1}{2}(\mathbf{L} - \mathbf{L}^T),$$

where we define the *rate of deformation* tensor as the symmetric part of $\mathbf{L}$

$$\mathbf{D} = \frac{1}{2}(\mathbf{L} + \mathbf{L}^T), \tag{3.17}$$

and we define the *spin*, or *vorticity*, tensor as the antisymmetric part of $\mathbf{L}$,

$$\mathbf{W} = \frac{1}{2}(\mathbf{L} - \mathbf{L}^T). \tag{3.18}$$

3.3 Stress Definitions

In small deformation theory, there is no distinction between a material, or reference configuration, and a spatial, or current configuration. In finite deformation kinematics, however, we must take these two configurations into consideration. Let us start by considering a traction acting on a differential area element in the current configuration. Let the traction, $\mathbf{t}$, be defined as a force acting on a surface in the current configuration per unit current configuration area. As such, let a differential resultant force in the current configuration be defined as,

$$\mathrm{df} = \mathbf{t}\mathrm{da} = \mathbf{T}\mathbf{n}\mathrm{da}.$$

Here, $\mathbf{T}$ is the Cauchy stress tensor and $\mathbf{n}$ is the outward unit normal to the current configuration surface with differential area, da. We want some way to relate the differential force in the current configuration to a differential area element in the reference configuration, dA, with corresponding outward unit normal, $\mathbf{N}$. Using the relation, $\mathbf{n}\mathrm{da} = J\mathbf{F}^{-1}\mathbf{N}\mathrm{dA}$ (i.e. Nanson's formula), where J = det($\mathbf{F}$), we obtain the following:

$$\mathbf{t}\mathrm{da} = \mathbf{T}\mathbf{n}\mathrm{da} = J\mathbf{T}\mathbf{F}^{-T}\mathbf{N}\mathrm{dA}. \tag{3.19}$$

From this relation we define the *first Piola Kirchhoff* stress tensor,

$$\mathbf{P} = J\mathbf{T}\mathbf{F}^{-\mathrm{T}}. \tag{3.20}$$

In component form,

$$P_{i\alpha} = JT_{ij}F^{-\mathrm{T}}_{j\alpha}. \tag{3.21}$$

This stress measure is the most convenient for evaluating our subsequent weak form integrals, which will appear in our finite element residual equations, in that all our finite element

shape function gradients will be with respect to the global reference configuration. This *total Lagrangian* formulation allows us to formulate our shape functions and gradients only *once*, which is convenient computationally.

3.4 Stress Rate and the Kinematic Update

We have briefly described the motion of a continuum between arbitrary reference and current configurations. Ultimately, however, we are going to be interested in advancing the motion from the beginning of a time step, at time t_n, to the end of a time step, at time t_{n+1}, while retaining a global reference configuration of the body at time t_0. As a result, we must elaborate our description of the kinematics of a continuum solid body in anticipation of a numerical implementation in the discrete time setting.

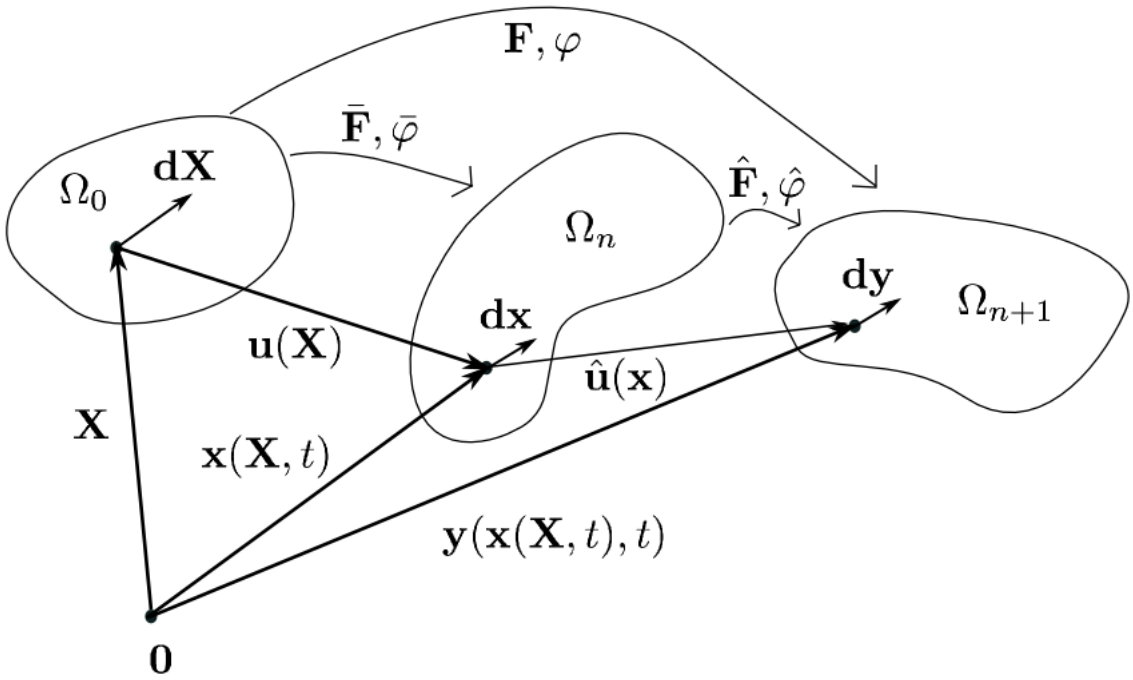

Figure 2: Representation of an arbitrary motion, depicting the reference configuration, the current, beginning-step configuration at time t_n, and the updated, end-step configuration at time t_{n+1}.

Figure 2 shows the arbitrary motion of a body, whose open domain is denoted as Ω, and whose closure is denoted as $\bar{\Omega}$. There are three configurations of the body shown. The first is denoted as Ω_0, and is the reference configuration at time $t_0 = 0$. The second configuration is denoted as Ω_n, and is the beginning-step configuration at time t_n. The third configuration is Ω_{n+1}, and is the end-step configuration at time t_{n+1}. The last two configurations represent the motion over an arbitrary timestep, where $\Delta t = t_{n+1} - t_n$ denotes the time step size. Furthermore, $\mathbf{X} \in \bar{\Omega}_0$ is the vector to a material point in the closure of Ω in the reference configuration. Similary, $\mathbf{x}$ and $\mathbf{y}$ represent analogous vectors in the beginning-step and end-step configurations, respectively.

The deformation between the reference configuration, Ω_0, and the end-step configuration, Ω_{n+1}, is described by the deformation gradient, $\mathbf{F}$, or the mapping $\varphi : \mathbf{X} \to \mathbf{y}$, as is sometimes convenient. The deformation between the reference configuration, Ω_0, and the beginning-step configuration, Ω_n, is described by the deformation gradient, $\bar{\mathbf{F}}$, or the mapping $\bar{\varphi} : \mathbf{X} \to \mathbf{x}$. Lastly, the deformation between the beginning-step configuration, Ω_n, and the end-step configuration, Ω_{n+1}, referred to here as the "incremental motion," is described by the deformation gradient, $\hat{\mathbf{F}}$, or the mapping $\hat{\varphi} : \mathbf{x} \to \mathbf{y}$.

The total motion of the body, described by $\mathbf{F}$, may be multiplicatively decomposed in terms of the other two motions as follows:

$$\mathbf{F} = \hat{\mathbf{F}}\bar{\mathbf{F}}. \tag{3.22}$$

From this decomposition, the incremental motion may be expressed as

$$\hat{\mathbf{F}} = \mathbf{F}\bar{\mathbf{F}}^{-1}. \tag{3.23}$$

Suppose that we know the total deformation as well as the beginning-step deformation. We can then obtain the incremental deformation gradient, $\hat{\mathbf{F}}$, via Equation 3.23. Our motivation to do so can be stated as follows:

Given an incremental motion described by the incremental deformation gradient, $\hat{\boldsymbol{F}}$, we want to perform the polar decomposition of $\hat{\boldsymbol{F}}$, that is, $\hat{\boldsymbol{F}} = \hat{\boldsymbol{R}}\hat{\boldsymbol{U}}$, in order to obtain an incremental stress update, where $\hat{\boldsymbol{R}}$ and $\hat{\boldsymbol{U}}$ are the rotation and stretch tensors associated with the incremental motion, respectively.

On a practical note, many constitutive update routines work with the rate form of the Cauchy stress tensor. Here we define one such rate, known as the *Jaumann* rate,

$$\mathring{\mathbf{T}} = \dot{\mathbf{T}} + \mathbf{TW} - \mathbf{WT}, \tag{3.24}$$

where $\mathbf{T}$ is the Cauchy stress tensor and $\mathbf{W}$ is the spin tensor. The actual constitutive update of $\mathbf{T}$ will depend on a stretch rate quantity associated with $\hat{\mathbf{U}}$ from the polar decomposition of $\hat{\mathbf{F}}$, and the forward rotation of that stress will depend on $\hat{\mathbf{R}}$ from the same polar decomposition. In the finite deformations case, we may use Equation 3.20 to then transform the updated Cauchy stress to the first Piola Kirchhoff stress for use in a weak equilibrium statement.

There are many ways to perform this incremental stress/kinematic update. For specifics of what has been used in this work, see [2].

3.5 Hyperelastic Constitutive Model

Per reference [3], a compressible Mooney-Rivlin material model is used to characterize hyperelastic material response. A hyperelastic constitutive model treats nonlinear elastic material behavior at finite strains. The material model can be expressed in terms of an energy density functional. For a compressible Mooney-Rivlin material model this takes the form

$$W = C_1(\bar{I}_1 - 3) + C_2(\bar{I}_2 - 3) + D_1(J-1)^2, \tag{3.25}$$

where W is the strain energy density, C_1 and C_2 are constants related to distortional response, and D_1 is a constant related to volumetric response; J is defined in Equation 3.8. The quantities $\bar{I}_1$ and $\bar{I}_2$ are defined as $\bar{I}_1 = J^{-\frac{2}{3}} I_1$ and $\bar{I}_2 = J^{-\frac{4}{3}} I_2$, where I_1 and I_2 are the first and second invariants of $\mathbf{B} = \mathbf{F}\mathbf{F}^T$. These invariants are functions of λ_i, where λ_i are the eigenvalues of the stretch part of the deformation gradient $\mathbf{F}$. Specifically, $I_1 = \lambda_1^2 + \lambda_2^2 + \lambda_3^2$ and $I_2 = \lambda_1^2\lambda_2^2 + \lambda_2^2\lambda_3^2 + \lambda_1^2\lambda_3^2 = \frac{1}{2}[(\mathrm{tr}\mathbf{B})^2 - \mathrm{tr}(\mathbf{B}^2)]$. Lastly, the relationship between the Cauchy stress $\mathbf{T}$, and the strain energy density W, is

$$\mathbf{T} = \frac{1}{J}\frac{\partial W}{\partial \mathbf{F}}\mathbf{F}^T. \tag{3.26}$$

A hyperelastic material model is used for various numerical examples presented in Section 11 where finite strains and/or rotations are experienced. For more details regarding this material model, including the full expression for Equation 3.26 in terms of the deformation gradient, see [3].

3.6 Hypoelastic Constitutive Model

A hypoelastic material model is expressed using a rate-form of the linear elasticity equations

$$\dot{T}_{ij} = C_{ijkl}\dot{\epsilon}_{kl}, \tag{3.27}$$

where $\mathbf{T}$ is the Cauchy stress tensor, $\mathbf{C}$ is the elastic modulus tensor, and ϵ is the infinitesimal strain tensor. The idea behind a hypoelastic material is to extend this rate form to finite strains, for which the inifinitesimal strain tensor in combination with the time rate of the Cauchy stress are not sufficient. In order to work around this, the following hypoelastic constitutive relation is presented:

$$\mathring{T}_{ij} = C_{ijkl}D_{kl}, \tag{3.28}$$

where $\mathring{\mathbf{T}}$ is the Jaumann rate of the Cauchy stress and $\mathbf{D}$ is the rate of deformation tensor.

For a finite element code that handles incremental kinematics, it is often convenient to implement a hypoelastic constitutive model following the rate equation in 3.28. For this reason, some of the numerical examples presented in Section 11 use a hypoelastic constitutive relation, which is suitable for the small strains and rotations experienced in those problems.

4 Finite Element Formulation in Nonlinear Solid Mechanics

4.1 Strong Form of the IBVP in Finite Strains

Let us introduce a single continuum body in the reference configuration whose open domain is denoted Ω_0, and whose closure is denoted by $\bar{\Omega}_0$. The boundary of Ω_0, denoted $\partial\Omega_0$, is composed of two regions. One region of the boundary is that over which traction boundary conditions are prescribed, and the other region is that over which displacement boundary conditions are prescribed. These regions are denoted Γ_t and Γ_u, respectively, such that $\Gamma_t \cup \Gamma_u = \partial\Omega_0$, and $\Gamma_t \cap \Gamma_u = \varnothing$.

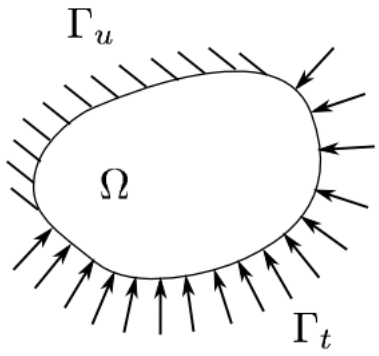

Figure 3: Representation of the body, Ω_0, in the reference configuration showing regions of specified displacement and traction boundary conditions, denoted Γ_u and Γ_t, respectively.

Figure 3 shows the aforementioned body and the regions over which displacement and traction boundary conditions are specified. In the following section, we will consider a linear momentum balance to derive the equations of motion governing the deformation of this body. We will distinguish, for clarity, the difference between integrals over the open domain of the body in the reference configuration and in the current configuration. The open domain of the

current configuration will be denoted as Ω, where we have dropped the subscript indicating a particular time step.

4.1.1 Equations of Motion

Let us consider the following momentum balance on an arbitrary subset of the reference configuration body, Ω_0,

$$\int_{\partial\omega_0} P_{i\alpha}N_\alpha dA + \int_{\omega_0} \rho_0 b_i dV = \frac{d}{dt}\int_{\omega_0} \rho_0 v_i dV, \tag{4.1}$$

where $\omega_0 \subset \Omega_0$ and $\partial\omega_0$ represents the boundary of ω_0. The first term on the left hand side of Equation 4.1 is the integral of the first Piola Kirchhoff traction vector, $P_{i\alpha}N_\alpha = \bar{p}_i$, acting on $\partial\omega$. The second term on the left hand side and the right hand side of Equation 4.1, are the integrals of the body force per unit reference volume and the time rate of change of linear momentum over the reference configaration volume, respectively. The vectors **b** and **v** are the prescribed body force per unit mass, and the material, or reference configuration, velocity field, respectively; and ρ_0 is the mass density per unit reference configuration volume. These integrals are derived by transforming the integral in the current configuration to an integral in the reference configuration. This is done with the following relation:

$$dv = JdV, \tag{4.2}$$

where dv is a volume element in the current configuration, dV is a volume element in the reference configuration, and J = det(**F**). The integral of the body force over the current configuration volume then transforms to an integral over the reference configuration volume as

$$\int_\omega \rho b_i dv = \int_{\omega_0} \rho b_i J dV = \int_{\omega_0} \rho_0 b_i dV, \tag{4.3}$$

where $\omega \subset \Omega$, ρ is the mass density per unit current volume, and $J\rho = \rho_0$. Furthermore, let us consider how the linear momentum term transforms from an integral over the current configuration volume to an integral over the reference configuration volume,

$$\frac{d}{dt}\int_{\omega} \rho v_i dv = \frac{d}{dt}\int_{\omega_0} \rho v_i J dV = \frac{d}{dt}\int_{\omega_0} \rho_0 v_i dV = \int_{\omega_0} \rho_0 a_i dV, \tag{4.4}$$

where **a** is the material, or reference configuration, acceleration field.

Lastly, we need to transform the area integral of the first term on the left hand side of Equation 4.1 to an integral over the reference configuration volume. After applying the divergence theorem to this term, this integral reads

$$\int_{\partial\omega_0} P_{i\alpha} N_\alpha dA = \int_{\omega_0} P_{i\alpha,\alpha} dV. \tag{4.5}$$

Putting Equations 4.3- 4.5 together leads to the following equation,

$$\int_{\omega_0} [P_{i\alpha,\alpha} + \rho_0 b_i - \rho_0 a_i] dV = 0. \tag{4.6}$$

Using the localization theorem, which states that in order for the integral in Equation 4.7 to hold over the body, Ω_0, the integrand must equal zero pointwise. As a result, we have our strong-form governing equations of motion:

$$P_{i\alpha,\alpha} + \rho_0 b_i - \rho_0 a_i = 0, \ \ \forall\, X_i \in \bar{\Omega}_0, t > 0. \tag{4.7}$$

4.1.2 Strong Form

The complete strong-form problem stateent reads as follows. For all t ∈ [0,T], T > 0, and all $\mathbf{X} \in \Omega_0$, find $\mathbf{u}(\mathbf{X},t)$ such that,

$$P_{i\alpha,\alpha} + \rho_0 b_i - \rho_0 a_i = 0, \;\; X_i \in \Omega_0, \; t \in [0,T], T > 0, \tag{4.8}$$

$$P_{i\alpha} N_\alpha = \bar{p}_i, \;\; \forall \, X_i \in \Gamma_t, \tag{4.9}$$

$$u_i = \bar{u}_i, \;\; \forall \, X_i \in \Gamma_u, \; t \in [0,T], T > 0, \tag{4.10}$$

$$\dot{u}_i \rfloor_{t=0} = \bar{v}_i, \;\; \forall \, X_i \in \bar{\Omega}_0, \tag{4.11}$$

$$u_i \rfloor_{t=0} = 0, \;\; \forall \, X_i \in \bar{\Omega}_0, \tag{4.12}$$

where t is time,

Ω_0 represents the open domain in the reference configuration,

$\bar{\Omega}_0$ represents the closure of domain, Ω_0, in the reference configuration

$\partial\Omega_0$ represents the boundary of the body, $\bar{\Omega}_0 \setminus \Omega_0$,

Γ_t represents the part of the boundary over which traction B.C.s are specified,

Γ_u represents the part of the boundary over which displacement B.C.s are specified,

$\Gamma_t \cap \Gamma_u = \varnothing$, $\Gamma_t \cup \Gamma_u = \partial\Omega_0$, and

$\bar{\mathbf{p}}$ is the prescribed first Piola Kirchoff traction vector.

4.2 Weak Form of the IBVP in Finite Strains

The finite element formulation for our solid mechanics problem involves taking the strong form of the equations of motion and deriving the weak, or variational, form. Let us recall

our governing equations of motion,

$$\nabla \cdot \mathbf{P} + \rho_0 \mathbf{b} = \rho_0 \mathbf{a}, \ \ \forall \ \mathbf{X} \in \Omega_0, t > 0. \tag{4.13}$$

To develop the weak form, we must consider the test or weighting space, $\mathcal{V}$ defined as follows,

$$\mathcal{V} = \left\{ \mathbf{w} \,|\mathbf{w} \in \ H^1(\Omega_0), \mathbf{w} = \mathbf{0} \ \text{ on } \Gamma_u \right\} \tag{4.14}$$

Note that a test function, $\mathbf{w}$, is entirely defined in the reference configuration. Furthermore, let us define the solution space $\mathcal{S}$ as follows,

$$\mathcal{S} = \left\{ \mathbf{u} \,|\mathbf{u} \in \ H^1(\Omega_0), \mathbf{u} = \overline{\mathbf{u}} \ \text{ on } \Gamma_u \right\}. \tag{4.15}$$

Note that this solution space holds for all time, t. With the definition of $\mathcal{V}$ and $\mathcal{S}$ in hand, we may derive the weak form of 4.13. To do so, we dot Equation 4.13 with a test function $\mathbf{w}$, and integrate over the body in the reference configuration.

$$\int_{\Omega_0} \nabla \cdot \mathbf{P} \cdot \mathbf{w} \, dV + \int_{\Omega_0} \rho_0 (\mathbf{b} - \mathbf{a}) \ \cdot \mathbf{w} \, dV = \mathbf{0}, \tag{4.16}$$

Equation 4.16 written in component form is

$$\int_{\Omega_0} P_{i\alpha,\alpha} w_i dV + \int_{\Omega_0} \rho_0 (b_i - a_i) w_i dV = 0. \tag{4.17}$$

Using the divergence theorem on the first term of the left hand side of Equation 4.17 yields the weak form of the IBVP:

Find $\mathbf{u}(\mathbf{X}, t) \in S$ such that,

$$\begin{aligned} &\int_{\Omega_0} P_{i\alpha} w_{i,\alpha} dV - \int_{\Omega_0} \rho_0 (b_i - a_i) w_i dV - \int_{\Gamma_t} \overline{p}_i w_i dA = 0, \ \forall \, \mathbf{w} \in V, \\ &\text{where,} \\ &\mathcal{V} = \left\{ \mathbf{w} \,|\mathbf{w} \in \ H^1(\Omega_0), \mathbf{w} = \mathbf{0} \ \text{ on } \Gamma_u \right\}, \\ &\mathcal{S} = \left\{ \mathbf{u} \,|\mathbf{u} \in \ H^1(\Omega_0), \mathbf{u} = \bar{\mathbf{u}} \ \text{ on } \Gamma_u \right\}. \end{aligned} \tag{4.18}$$

4.3 Introduction to Galerkin Approximations

The finite element approximation is a displacement based Galerkin approximation where a solution is sought in a finite dimensional subspace of the infinite dimensional solution space, $\mathcal{S}$, appearing in Equation 4.18. This is accomplished by discretizing the body's reference configuration, Ω_0, into "elements", where each element contains a specific number of nodes. The spatial discretization allows us to formulate a discrete version of the weak form of our IBVP. Before we do this, let us introduce some key concepts regarding Galerkin approximations, as applied to the finite element solution of the weak form of our IBVP.

4.3.1 Finite Dimensional Approximation Spaces

We start by writing the Galerkin approximations to $\mathbf{u}$ and $\mathbf{w}$ in terms of a finite set of basis functions, $\phi_a(\mathbf{X})$, associated with nodes, a, in the discretized body. The approximations are as follows:

$$\mathbf{w}^h(\mathbf{X}, t) = \sum_{a \in \mathcal{A}} \mathbf{w}_a \phi_a(\mathbf{X}), \tag{4.19}$$

and

$$\mathbf{u}^h(\mathbf{X}, t) = \sum_{a \in \mathcal{A}} \mathbf{u}_a \phi_a(\mathbf{X}), \tag{4.20}$$

where the superscript h denotes the discrete approximation to the trial solution and weighting function, $\mathbf{u}$ and $\mathbf{w}$, respectively, and $\mathcal{A}$ is the set of all nodes excluding those with prescribed displacement boundary conditions for which the variation, $\mathbf{w}$, is equal to zero. Furthermore, $\phi_a(\mathbf{X})$ are continuous (at least C^0) basis functions defined for all $\mathbf{X} \in \Omega_0$, and $\mathbf{u}_a$ and $\mathbf{w}_a$ are the vectors of trial solution and weighting coefficients associated with nodes a $\in \mathcal{A}$. The finite dimensional approximations to $\mathcal{V}$ and to $\mathcal{S}$, denoted $\mathcal{V}^h$ and $\mathcal{S}^h$, are obtained by plugging 4.19 and 4.20 into 4.14 and 4.15, respectively. Note that in the Galerkin finite element method, the same basis functions are used in the finite dimensional approximations to $\mathcal{V}$ and to $\mathcal{S}$. Note that $\mathbf{u}^h \approx \bar{\mathbf{u}}$ in the sense that the discrete representation of Γ_u is an approximation to the geometry of Γ_u in the continuous setting. There are a few key properties of the finite element basis functions, $\phi_a(\mathbf{X})$, that are discussed next.

4.3.2 Compact Support

Compact support refers to the notion that the basis functions, $\phi_a(\mathbf{X})$, are nonzero only for material points that lie in elements that contain node a. That is,

$$\phi_a(\mathbf{X}) = 0 \; \forall \; \mathbf{X} \notin \mathcal{I}, \tag{4.21}$$

where $\mathcal{I}$ denotes the set of all elements that contain node a.

Compact support is important in the traditional finite element method for a couple of reasons. First, the resulting global stiffness matrix is banded and sparse. Second, compact support leads to a computationally convenient element-by-element calculation and assembly of contributions to the global residual. These facts facilitate an ease in the algorithmization of the method, and allow for efficient computation of solutions.

4.3.3 Partition of Unity

Partition of unity goes hand-in-hand with compact support to allow us to break up our globally defined, continuous basis functions into shape functions defined locally over elements. The partition of unity states that the sum of the basis functions over the nodes of an element equal unity. That is,

$$\sum_a \phi_a(\mathbf{X}) = 1. \tag{4.22}$$

This characteristic of the basis functions will be used in Section 4.3.5.

4.3.4 Kronecker Delta Property

The Kronecker Delta property essentially states that 4.20 is an interpolation of the nodal data. It can be stated as

$$\phi_a(\mathbf{X}_b) = \delta_{ab}. \tag{4.23}$$

The Kronecker Delta property is significant in that it allows for the convenient enforcement of displacement boundary conditions. Without this property, the enforcement of essential boundary conditions may become problematic.

4.3.5 Consistency and Completeness

Consistency and completeness are the two conditions required to show convergence of the finite element method. Consistency, which will not be explicitly shown here, refers to the consistency of the stress divergence term (4.17) as it is actually computed in the approximate problem with the stress divergence term as it actually occurs in the original variational boundary value problem (i.e. with exact integration).

Completeness, also referred to as approximability or reproducability, is the requirement that the Galerkin approximation be capable of exactly reproducing a polynomial up to a certain order. This order is one for the weak form of the initial boundary value problem presented in this work. If the nodal values are set consistent with a global linear function in three dimensions, for example, the Galerkin approximation ought to reproduce this linear field. To explore this, let us define a linear field as follows:

$$u(\mathbf{x}) = a_0 + a_1 x_1 + a_2 x_2 + a_3 x_3. \tag{4.24}$$

The finite element interpolant is then

$$\begin{aligned} \sum_{a=1}^{N} u_a \phi_a &= \sum_{a=1}^{N} (a_0 + a_1 x_{1a} + a_2 x_{2a} + a_3 x_{3a}) \phi_a \\ &= a_0 \sum_{a=1}^{N} \phi_a + a_1 \sum_{a=1}^{N} x_{1a} \phi_a + a_2 \sum_{a=1}^{N} x_{2a} \phi_a + \ a_2 \sum_{a=1}^{N} x_{3a} \phi_a \\ &= a_0 1 + a_1 x_1 + a_2 x_2 + a_3 x_3, \ \forall \ a_0, a_1, a_2, a_3, \end{aligned} \tag{4.25}$$

where N is the number of nodes in the mesh. The partition of unity requirement give us the first term on the third line of equation 4.25. In order to reproduce the linear field, however, the following requirement must hold

$$x_i = \sum_{a=1}^{N} x_{ia} \phi_a. \tag{4.26}$$

Provided equation 4.26 is satisfied by the basis functions, then equation 4.25 results in the final line, which is in fact the original linear field. Equation 4.26 is a requirement on the basis functions themselves. As a note, linear isoparametric shape functions exhibit 4.26 by construction.

For practical purposes, a finite element approximation exhibits convergence if the approx-

imate solution tends toward the exact solution as the mesh size decreases (i.e. increase the finite dimensional approximation space). Convergence is often demonstrated by mesh refinement studies and patch tests.

4.4 Galerkin Approximation and the Discrete Equations of Motion

We may derive the discrete equations of motion by plugging the discrete Galerkin approximations shown in Equations 4.19 and 4.20 into Equation 4.18 to obtain

$$\begin{aligned}&\int_{\Omega_0} P_{i\alpha} \sum_{b\in\mathcal{A}} w_{ib}\phi_{b,\alpha} dV - \int_{\Omega_0} \rho_0 (b_i - \sum_{a\in\mathcal{A}} \ddot{u}_{ia}\phi_a) \sum_{b\in\mathcal{A}} w_{ib}\phi_b dV - \\ &\int_{\Gamma_t} \bar{p}_i \sum_{b\in\mathcal{A}} w_{ib}\phi_b dA = 0.\end{aligned} \tag{4.27}$$

where $\ddot{\mathbf{u}}$ is the material acceleration field. In this case, $\ddot{\mathbf{u}} = \mathbf{a}$, where the former is more common in the structural dynamics literature, and the latter is a natural notation selection when denoting the accleration field of the body in the reference configuration when deriving the linear momentum balance shown in Equation 4.1. Equation 4.27 can be rearranged as follows,

$$\begin{aligned}&\sum_{b\in\mathcal{A}} w_{ib} \int_{\Omega_0} P_{i\alpha}\phi_{b,\alpha} dV - \sum_{b\in\mathcal{A}} w_{ib} \int_{\Omega_0} \rho_0 b_i \phi_b dV \\ &+ \sum_{b\in\mathcal{A}}^{N} w_{ib} \int_{\Gamma_t} \bar{p}_i \phi_b dA + \sum_{a\in\mathcal{A}} \sum_{b\in\mathcal{A}} w_{ib} \int_{\Omega_0} (\rho_0 \phi_a \delta_{i\alpha} \phi_b) \ddot{u}_{\alpha a} dV = 0.\end{aligned} \tag{4.28}$$

Let us define in component form, the mass matrix,

$$M_{ia\alpha b} = \rho_0 \int_{\Omega_0} \phi_a \delta_{i\alpha} \phi_b dV, \tag{4.29}$$

the external force vector,

$$F_{ib}^{ext} = \int_{\Omega_0} \rho_0 B_i \phi_b dV + \int_{\Gamma_t} \overline{p}_i \phi_b dA, \tag{4.30}$$

and the internal force vector,

$$F_{ib}^{int} = \int_{\Omega_0} P_{i\alpha} \phi_{b,\alpha} dV. \tag{4.31}$$

Equations 4.29 - 4.31 can be combined with Equation 4.28 to form the following expression,

$$\sum_{a \in \mathcal{A}} w_{ia} \left(F_{ia}^{int} + M_{ia\alpha b} \ddot{u}_{\alpha a} - F_{ia}^{ext} \right) = 0. \tag{4.32}$$

Because the w_{ia} are arbitrary variations, the following must hold:

$$F_{ia}^{int} + M_{ia\alpha b} \ddot{u}_{\alpha a} - F_{ia}^{ext} = 0. \tag{4.33}$$

Equation 4.33 is the finite element *residual* and may be rewritten In direct notation as,

$$\mathbf{R}_a = \mathbf{F}_a^{int} + \mathbf{M}\ddot{\mathbf{u}}_a - \mathbf{F}_a^{ext} = \mathbf{0}, \tag{4.34}$$

and where a ranges over all unconstrained nodes in the mesh.

4.5 Finite Elements and the Element-Wise Construction of the Global Equations

The integral terms in Equation 4.34, i.e. Equations 4.29 - 4.31, are computed on an element-by-element basis. This feature of the discretization allows us to replace the basis functions defined for all $\mathbf{X} \in \Omega_0$ with shape functions defined locally on each element. Let N_a denote the shape function associated with node a of a single element. Then, Equation 4.34 can

be expanded and rewritten where each integral is the sum of integrals over elements in the discretized body:

$$\sum_{m=1}^{M}\left[\int_{\Omega_m} P_{i\alpha}N^m_{b,\alpha}dV + \rho_0\int_{\Omega_m} N^m_a\delta_{i\alpha}N^m_b dV\ddot{u}_{\alpha a} - \int_{\Omega_m}\rho_0 b_i N^m_b dV + \int_{\partial_t\Omega_m}\bar{p}_i N^m_b dA\right] = 0, \tag{4.35}$$

where $\Omega_m \subset \Omega$ is the domain occupied by element m, M is the total number of elements in the mesh, and a, b = 1, 2,...,n, where n is the number of nodes in the element, and $\partial_t\Omega_m$ is the boundary of element m over which traction boundary conditions are specified. To be clear, Equation 4.35 is a sum over element contributions to the global set of equations at a single node a. The complete equation at node a is composed of the sum, or assembly of all element contributions from elements that share node a. The fact that the finite element basis functions have compact support is what allows us to compute element level calculations (i.e. Equation 4.35) in this way and then assemble their contributions into the global system of equations. As a note, rather than run through all nodes a and/or b that appear in each of the element integrals in Equation 4.35, we restrict a and b to range only over the nodes of element m. This will be implied throughout where element level integrals appear.

In order to compute the integrals in Equation 4.35, we use Gaussian quadrature. To do so, we map the element in the physical reference configuration to a canonical element in parent space such that, for $\mathbf{X} \in \Omega_m$, there exists a $(\xi, \eta, \zeta) \in [-1, 1]^3$ in the parent space. Additionally, the shape functions associated with each node a are defined in the parent space. That is, $N_a = N_a(\xi, \eta, \zeta) : \xi, \eta, \zeta \in \Omega_e$, where Ω_e denotes the canonical element (i.e. in the parent space). This is known as the isoparametric mapping.

4.5.1 Isoparametric Mapping

The isoparametric mapping is one-to-one and onto and allows us to map an element in physical space to a canonical element in parent space. This is performed using the isoparametric mapping:

$$x_i = \sum_{a}^{n} x_{ia} N_a(\xi, \eta, \zeta), \tag{4.36}$$

where n is the total number of nodes belonging to an element and N_a is the shape function associated with node a of that element. The parent space serves as the domain over which the shape functions are defined. The linear combination in Equation 4.36 maps a (ξ, η, ζ) coordinate point in parent space to a coordinate point (x_1, x_2, x_3) in physical space, where $x_{i\alpha}$ are the coordinates of node α in physical space and (ξ, η, ζ) are the three coordinate directions in parent space. The parent space additionally serves to define the quadrature rule with which numerical integration of the finite element equations is performed. Usually, the shape functions are low order polynomials suitable for Gaussian quadrature. In order to evaluate these integrals, they must be transformed from integrals over the physical element to integrals over the parent element. To demonstrate how this is done, let us focus on the stress-divergence term of Equation 4.35, which is the first term on the right hand side. The term is,

$$\int_{\Omega_m} P_{i\alpha} N_{b,\alpha} dV. \tag{4.37}$$

The crux of the isoparametric transformation is two fold: the transformation of the domain of integration, and the computation of derivatives of shape functions, which are defined in parent space, with respect to physical coordinates.

For the transformation of the volume, let us define the Jacobian of the transformation in 2D

(the extension to 3D is straightforward and follows the steps outlined here), as

$$\mathbf{J} = \begin{bmatrix} x_{1,\xi} & x_{1,\eta} \\ x_{2,\xi} & x_{2,\eta} \end{bmatrix} \tag{4.38}$$

Though not proven here, the volume element transforms as

$$dV = (\text{det}\mathbf{J})d\xi d\eta, \tag{4.39}$$

where we define $J = \text{det}\mathbf{J}$. The integral therefore transforms as

$$\int_{\Omega_m} f dV = \int_{\Omega_e} f(\text{det}\mathbf{J})d\xi d\eta, \tag{4.40}$$

where Ω_e denotes the open domain of the canonical element.

For the computation of the derivatives of the element shape functions with respect to the physical coordinates (over which they are not explicitly defined), let us expand the derivative using the chain rule as

$$\frac{\partial N_a}{\partial x_i} = N_{a,\xi}\xi_{,i} + N_{a,\eta}\eta_{,i}. \tag{4.41}$$

In matrix-vector notation, Equation 4.41 may be written

$$\begin{bmatrix} N_{a,1} \\ N_{a,2} \end{bmatrix} = \begin{bmatrix} \xi_{,1} & \eta_{,1} \\ \xi_{,2} & \eta_{,2} \end{bmatrix} \begin{bmatrix} N_{a,\xi} \\ N_{a,\eta}. \end{bmatrix} \tag{4.42}$$

The derivatives of the shape functions with respect to parent coordinates (i.e. right hand side vector) is easily computed, whereas the derivatives of the parent coordinates with respect to the spatial coordinates is less clear. However, these derivatives can be shown to be related

to the Jacobian of the transformation, $\mathbf{J}$. Namely, it can be shown that

$$\mathbf{J}^{-T} = \begin{bmatrix} \xi_{,1} & \eta_{,1} \\ \xi_{,2} & \eta_{,2}. \end{bmatrix} \tag{4.43}$$

Thus,

$$\begin{bmatrix} N_{a,1} \\ N_{a,2} \end{bmatrix} = \mathbf{J}^{-T} \begin{bmatrix} N_{a,\xi} \\ N_{a,\eta}. \end{bmatrix} \tag{4.44}$$

It follows that Equation 4.37 transforms (in the two-dimensional case) as follows:

$$\int_{\Omega_m} P_{i\alpha} N_{b,\alpha} dV = \int_{\Omega_e} P_{i\alpha} N_{b,\alpha} J d\xi d\eta. \tag{4.45}$$

Note that all weak form integrals are over the element in the reference configuration. As a result, the shape function gradients, $N_{a,\alpha}(\xi, \eta, \zeta)$ are with respect to $X_\alpha, \alpha = 1, 2, 3$, in the reference configuration. These quantities need only be computed once, up front, for each element in the mesh in the reference configuration. This is a *total* Lagrangian formulation.

4.5.2 Eight Node Hex Element

This work presents a full 3-dimensional treatment of contact using trilinear, eight node hexadrons. These elements have eight nodes, and map to (ξ, η, ζ) space via the forgoing isoparametrc transformation where $2 \times 2 \times 2$ Gauss quadrature is used (see Section 4.5.3). Figure 4 shows a hex8 element in physical (x, y, z) space "mapped" to (ξ, η, ζ) parent space. The eight nodes are numbered counterclockwise starting at the bottom as shown in the figure.

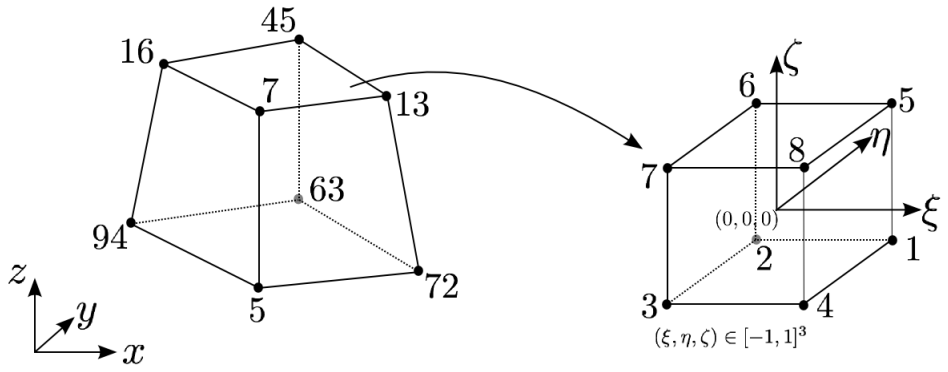

Figure 4: Physical configuration of the eight node hex element (on the left) mapped to parent element space. The parent element coordinates, (ξ, η, ζ), define the canonical eight node hex element.

The shape functions, defined in parent space, are defined as

$$N_i(\xi, \eta, \zeta) = \frac{1}{8}(1 + \xi\xi_i)(1 + \eta\eta_i)(1 + \zeta\zeta_i), \tag{4.46}$$

where (ξ_i, η_i, ζ_i) are the parent coordinates of the i^{th} node.

4.5.3 Gaussian Quadrature

The finite element method uses Gaussian quadrature to evaluate the element level integrals in the parent element space. In this work, $2 \times 2 \times 2$ Gauss quadrature is used where the integration points are defined in the parent space of the hex8 element. This three-dimensional quadrature rule is defined by

$$\int_{-1}^{1}\int_{-1}^{1}\int_{-1}^{1} f(\xi,\eta,\zeta)d\xi d\eta d\zeta \;\approx \sum_{k=1}^{nsp} w_k f(\xi_k,\eta_k,\zeta_k), \tag{4.47}$$

where nsp is the number of sampling points (i.e. integration points), (ξ_k, η_k, ζ_k) are the roots of the Legendre polynomial of order nsp, and w_k are the associated weights. For $2 \times 2 \times 2$ Gauss quadrature on a hex8 element, the parent space coordinates of the integration points are $\frac{1}{\sqrt{3}} * (\xi_i, \eta_i, \zeta_i)$, where (ξ_i, η_i, ζ_i) are the parent space coordinates of the i^{th} node. It can be shown that Gaussian quadrature can exactly evaluate polynomials up to order ($2 \times$ nsp - 1).

5 Nonlinear Solution Strategies

The solution of our nonlinear initial boundary value problem involves two main elements. We need a way to integrate our equations of motion, i.e. Equation 4.34, in time to obtain a set of global residual equations. We must then seek a displacement solution vector that satisfies this vector valued residual function. In this work, we use the HHT-alpha method for integrating our equations of motion in time, and we use Newton's Method to force our global residual vector to zero. The next sections outline the details of these two algorithms.

5.1 Time Integration Schemes

The HHT-alpha method [4] is used in this work for the time integration of the equations of motion. The reason this method was chosen in lieu of the Newmark family of methods [5], for example, is for its second order accuracy and the ability to introduce algorithmic damping. Let us start by introducing the equations used in the HHT method.

5.1.1 HHT Algorithm

Let us introduce the HHT algorithm by equations the equations of motion, the update equations, and the equations for the numerical constants. In the following equations, $\mathbf{d}_n$ and $\mathbf{d}_{n+1}$ are the beginning-step and end-step displacement vectors, respectively; $\mathbf{v}_n$ and $\mathbf{v}_{n+1}$ are the beginning-step and end-step velocity vectors, and $\mathbf{a}_n$ and $\mathbf{a}_{n+1}$ are the beginning-step and end-step acceleration vectors. The equations of motion in the discrete time setting is written

$$\mathbf{M}\mathbf{a}_{n+1} + (1+\alpha)\mathbf{K}\mathbf{d}_{n+1} - \alpha\mathbf{K}\mathbf{d}_n = (1+\alpha)\mathbf{F}_{n+1} - \alpha\mathbf{F}_n, \tag{5.1}$$

where α is an adjustable algorithmic parameter. The notation here is consistent with the structural dynamics literature and [4], where $\mathbf{K}$ is the stiffness matrix and $\mathbf{F}$ is the external forcing vector. The update for the end-step displacement vector is

$$\mathbf{d}_{n+1} = \mathbf{d}_n + \Delta t\mathbf{v}_n + \Delta t^2\left[(\frac{1}{2}-\beta)\mathbf{a}_n + \beta\mathbf{a}_{n+1}\right], \tag{5.2}$$

while the update for the end-step velocity vector is

$$\mathbf{v}_{n+1} = \mathbf{v}_n + \Delta t\left[(1-\gamma)\mathbf{a}_n + \gamma\mathbf{a}_{n+1}\right]. \tag{5.3}$$

Here, β and γ are two additional algorithmic parameters. The initial vectors at $t = 0$ are as follows:

$$\begin{aligned} \mathbf{d}_0 &= \bar{\mathbf{d}}, \\ \mathbf{v}_0 &= \bar{\mathbf{v}}, \\ \mathbf{a}_0 &= \mathbf{M}^{-1}(\mathbf{F}_0 - \mathbf{K}\mathbf{d}_0), \end{aligned} \tag{5.4}$$

where $\bar{\mathbf{d}}$ and $\bar{\mathbf{v}}$ are understood to be the prescribed initial displacement and velocity vectors. The expressions for the numerical constants present in Equations 5.1 - 5.3 are as follows:

$$\begin{aligned} \gamma &= \frac{1-2\alpha}{2}, \\ \beta &= \frac{(1-\alpha)^2}{4}, \\ \alpha &\in [-\frac{1}{3}, 0]. \end{aligned} \tag{5.5}$$

These parameters are chosen per [4] and are chosen to provide a second-order accurate method with algorithmic damping. The goal is to use Equations 5.1- 5.5 to obtain an expression for the finite element residual. To do so, we use Equation 5.2 to solve for $\mathbf{a}_{n+1}$. This yields the expression

$$\mathbf{a}_{n+1} = \frac{1}{\beta \Delta t^2} \left[\mathbf{d}_{n+1} - \mathbf{d}_n - \Delta t \mathbf{v}_n - \Delta t^2 (\frac{1}{2} - \beta) \mathbf{a}_n \right]. \tag{5.6}$$

We then substitute this expression into the expression for the end-step velocity, Equation 5.3, to obtain the following result for the end-step velocity solely in terms of the end-step and beginning-step displacement vectors and the beginning-step acceleration vector:

$$\mathbf{v}_{n+1} = \frac{\gamma}{\beta \Delta t}(\mathbf{d}_{n+1} - \mathbf{d}_n) + (1 - \frac{\gamma}{\beta})\mathbf{v}_n + \Delta t (1 - \frac{\gamma}{2\beta})\mathbf{a}_n. \tag{5.7}$$

Finally we plug Equation 5.6 and Equation 5.7 into Equation 5.1. Grouping terms and defining the incremental displacement, $\hat{\mathbf{d}} = \mathbf{d}_{n+1} - \mathbf{d}_n$, we arrive at the following expression for our finite element residual equation:

$$\begin{aligned}\mathbf{R} =& \mathbf{M}(\frac{1}{\beta\Delta t^2})\hat{\mathbf{d}} + (1+\alpha)\mathbf{K}\mathbf{d}_{n+1} \\ &- \alpha\mathbf{K}\mathbf{d}_n - \mathbf{M}(\frac{1}{\beta\Delta t})\mathbf{v}_n - \mathbf{M}(\frac{1}{2\beta} - 1)\mathbf{a}_n \\ &- [(1+\alpha)\mathbf{F}_{n+1} - \alpha\mathbf{F}_n] = \mathbf{0}.\end{aligned} \tag{5.8}$$

The global-length residual vector in Equation 5.8 is a function of the unknown vector $\mathbf{d}$. The residual is written per [4] without contact terms. Keep in mind that there are traction unknowns for contact problems. Thus, $\mathbf{d}$ would represent the global-length unknown vector *including* these tractions. The residual in Equation 5.8 is *linear* as written, but can be modified for nonlinear problems, as would be the case for contact with finite deformations. That is, the linear stiffness term, $\mathbf{K}\mathbf{d}_{n+1}$, would be replaced by a nonlinear, end-step stress divergence term. Additional contact terms would be included that are functions of the end-step and beginning-step contact traction portion of the modified $\mathbf{d}_{n+1}$ and $\mathbf{d}_n$ vectors, respectively. The approximate solution given a nonlinear residual is obtained via an implicit solution method. The nonlinear solution strategy used in this work to solve/enforce the residual equation is Newton's method, which is presented in the next section.

5.2 Newton's Method and the Finite Element Residual

Let us derive Newton's Method by considering the first order Taylor expansion of a vector valued function $\mathbf{f}(\mathbf{x})$, $\mathbf{x} \in \mathbb{R}^n$, about $\mathbf{x}^k$ evaluated at $\mathbf{x}^{k+1}$, i.e.

$$\mathbf{f}(\mathbf{x}) = \mathbf{f}(\mathbf{x}^k) + [D\mathbf{f}(\mathbf{x}^k)](\mathbf{x}^{k+1} - \mathbf{x}^k) = \mathbf{0}, \tag{5.9}$$

where $D\mathbf{f}(\mathbf{x}^k) = \frac{\partial \mathbf{f}(\mathbf{x}^k)}{\partial \mathbf{x}}$. Equation 5.9 can be rearranged to form an expression for the next solution iterate, $\mathbf{x}^{k+1}$,

$$\mathbf{x}^{k+1} = \mathbf{x}^k - D\mathbf{f}(\mathbf{x}^k)^{-1}\mathbf{f}(\mathbf{x}^k). \tag{5.10}$$

The finite element method seeks an incremental solution that advances the deformation from one timestep to the next. That is, we know the beginning step displacement solution, $\mathbf{u}_n$, at time t_n, and we seek an end step solution, $\mathbf{u}_{n+1}$ at time t_{n+1}, where $\hat{\mathbf{u}} = (\mathbf{u}_{n+1} - \mathbf{u}_n)$ is the incremental nodal displacement vector between time increments. Note that $\mathbf{u}$ will be used to denote the problem unknown, whereas $\mathbf{d}$ was used for the displacement unknown when presenting the HHT algorithm. This is done to stay more closely aligned with the notation used in [4]. Within this increment, the deformation and/or material response may be nonlinear. To handle this, we use Newton's method and linearize the residual about $\hat{\mathbf{u}}^k$ evaluated at some $\hat{\mathbf{u}}^{k+1}$ per Equation 5.9, where $\mathbf{f} = \mathbf{R}$ is taken to be the finite element residual. Let $\hat{\mathbf{u}}^k_{n+1}$ be the k^{th} Newton iterate for the incremental nodal displacement solution at time t_{n+1}. Within this notation, Equation 5.10 may be written as

$$D\mathbf{R}(\hat{\mathbf{u}}^k_{n+1})(\hat{\mathbf{u}}^{k+1}_{n+1} - \hat{\mathbf{u}}^k_{n+1}) = \ -\mathbf{R}(\hat{\mathbf{u}}^k_{n+1}). \tag{5.11}$$

Further, we write $\delta\hat{\mathbf{u}}_{n+1} = \hat{\mathbf{u}}^{k+1}_{n+1} - \hat{\mathbf{u}}^k_{n+1}$ as the correction to the incremental nodal displacement solution. Equation 5.11 may therefore be written

$$D\mathbf{R}(\hat{\mathbf{u}}^k_{n+1})\delta\hat{\mathbf{u}}_{n+1} = -\mathbf{R}(\hat{\mathbf{u}}^k_{n+1}). \tag{5.12}$$

Equation 5.12 is the linear system of equations whose solution provides an update to the incremental nodal displacement, advancing the solution from one time step to the next. Particular code implementation details, including solver types, are deferred until later in this work, when numerical implementations and numerical examples are presented.

6 Introduction to the Continuum Description of Large Deformation Contact

"Contact mechanics" implies the mechanical interaction of two or more bodies over a contact interface defined as the intersection of the boundaries associated with each body. For our purposes, we will only consider the interaction between two bodies, with the idea that characterizing the interaction between more than two bodies is a straightforward extension of the two body case. Furthermore, we will consider both bodies deformable, where deformable-to-rigid body contact is a special case that will not be explicitly addressed in this work.

6.1 Problem Preliminaries

To set up the continuous problem, let us consider an initial kinematic description of two deformable bodies. Figure 5 shows two deformable bodies in two different configurations. One is the reference configuration, with domains denoted as $\Omega_0^{(i)}, i = 1, 2$, where the subscript refers to the reference configuration and the superscript to body (1) or body (2), with i always ranging from 1 to 2 in this work. Furthermore, let us assume that each body undergoes a deformation process, or mapping, to the current configuration. This mapping is denoted as $\varphi^{(i)}$, and may be described in terms of the deformation gradients, $\mathbf{F}^{(i)}$, associated with each body.

The bodies in the current configuration are denoted by $\Omega_t^{(i)}$, where the subscript denotes the current time, $t \in [0, T], T > 0$. Further, let $\mathbf{X}^{(i)}$ and $\mathbf{x}^{(i)}$ be vectors associated with material points in the reference and current configurations, respectively. The surface normals are given by $\mathbf{N}^{(i)}$ and $\mathbf{n}^{(i)}$ in the reference and current configurations, respectively. Lastly, the

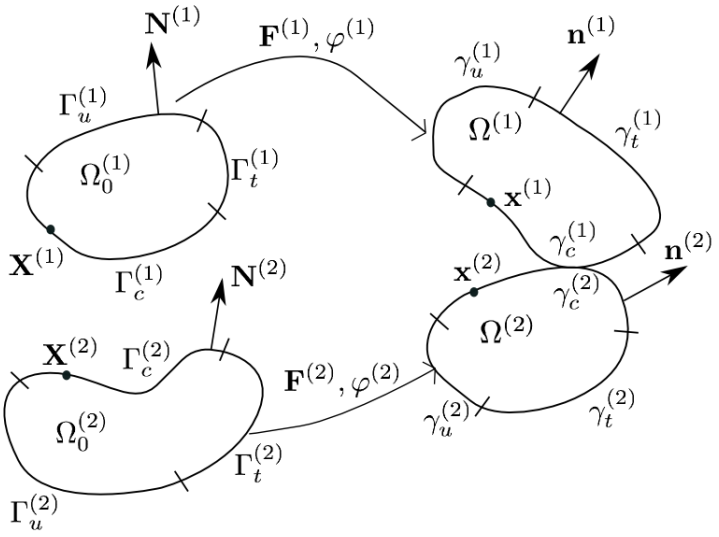

Figure 5: Reference and current configurations of two arbitrary deformable bodies. Noted on the boundary of each body are the portions over which displacement, traction, and contact boundary conditions are specified. This figure forms the basis for our two body, large deformation contact problem.

boundary of each body in the reference configuration is decomposed into portions over which displacement, traction, and contact boundary conditions are specified. That is, $\partial\Omega_0^{(i)} = \Gamma_u \cup \Gamma_t \cup \Gamma_c$ and $\bar{\Omega}_0^{(i)} \setminus \Omega_0^{(i)} = \partial\Omega_0^{(i)}$. Similarly, these regions map to the current configuration and are denoted as $\partial\Omega_t^{(i)} = \gamma_u \cup \gamma_t \cup \gamma_c$ and $\bar{\Omega}_t^{(i)} \setminus \Omega_t^{(i)} = \partial\Omega_t^{(i)}$.

6.2 Gap Function and the Contact Constraint

Imagine we have a current configuration point that lies on the portion of the boundary of body (1) over which contact boundary conditions are specified, and this point falls inside body (2) after some deformation process. That is, $\mathbf{x}_c^{(1)} \in \Omega_t^{(2)}$. This is clearly non-physical, and motivates the need to formulate a constraint that states that all points on one contact boundary surface *cannot* lie inside the other body. Another way to state this is that all points on one contact boundary surface are restricted to lie outside the closure of the other body, except that it may lie strictly *on* the contact boundary of that other body. That is,

contact occurs when $\mathbf{x}_c^{(1)} \in \gamma_c^{(1)}$, where $\gamma_c^{(1)} \cap \gamma_c^{(2)}$ is the portion of the two contact boundaries that are actually in contact, and $\gamma_c^{(1)} \cap (\partial\Omega_t^{(2)} \setminus \gamma_c^{(2)}) = \varnothing$ and $\gamma_c^{(1)} \cap \Omega_t^{(2)} = \varnothing$.

Note that the above implicitly assumes that the regions over which traction and displacement boundary conditions are specified cannot interact with a contact boundary on the other body. Furthermore, we are assuming that, while it is true that a point that lies *inside* body (1) *cannot* fall inside body (2), by enforcing zero interpenetration on the contact boundaries, we are implicitly precluding such events.

In order to enforce these restrictions, let us consider the aforementioned scenario where a point on the contact boundary of body (1) lies inside body (2). Figure 6 shows this scenario with the point denoted $\mathbf{x}_c^{(1)}$.

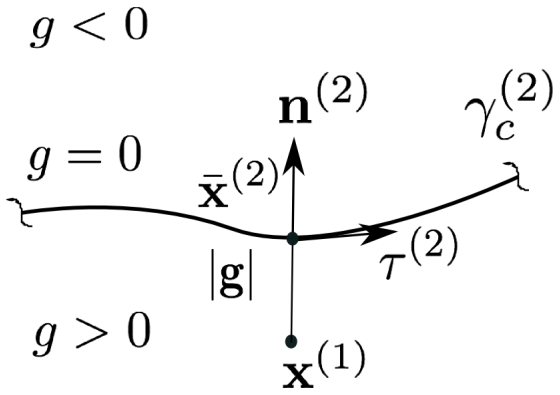

Figure 6: The normal gap definition per the closest point projection method. The slave point, $\mathbf{x}^{(1)}$, is shown with the associated closest point projection onto the master surface, $\bar{\mathbf{x}}^{(2)}$, which defines the gap and the associated magnitude of the gap, $|g|$, as shown in the figure. The sign of the gap function is shown corresponding to the exterior or the interior of the master body, with a value of zero for all slave points on the master surface, $\gamma_c^{(2)}$.

The discrete contact constraint must enforce either that body (1) cannot interpenetrate body (2), or vice-versa. For this presentation, the contact constraint will be formulated by

constraining $\gamma_c^{(1)}$ to lie on or outside $\gamma_c^{(2)}$[1]. Let us then consider the *closest point projection* of a point, $\mathbf{x}_c^{(1)}$, onto $\gamma_c^{(2)}$. The projected point, denoted $\bar{\mathbf{x}}_c^{(2)}$, is the closest point on $\gamma_c^{(2)}$ to $\mathbf{x}_c^{(1)}$. Formally, we may write this point as,

$$\bar{\mathbf{x}}_c^{(2)}(\mathbf{x}^{(1)}, t) = \arg \min_{\mathbf{x}^{(2)} \in \gamma_c^{(2)}} \|\mathbf{x}^{(1)} - \mathbf{x}^{(2)}\| \; \forall \, \mathbf{x}^{(1)} \in \gamma_c^{(1)}. \tag{6.1}$$

Equation 6.1 may be described by stating that for a given $\mathbf{x}^{(1)}$, find the point, $\bar{\mathbf{x}}_c^{(2)}$ on the surface $\gamma_c^{(2)}$ that yields the minimal distance from $\mathbf{x}^{(1)}$ to $\gamma_c^{(2)}$. Given the point that satisfies Equation 6.1, we may write a scalar valued gap function giving the magnitude of this minimal distance:

$$g(\mathbf{x}^{(1)}, t) = -\mathbf{n}^{(2)} \cdot (\mathbf{x}_c^{(1)} - \bar{\mathbf{x}}_c^{(2)}), \tag{6.2}$$

where $\mathbf{n}^{(2)}$ in Figure 6) is the outward facing normal to the surface, $\gamma_c^{(2)}$ at the point, $\bar{\mathbf{x}}_c^{(2)}$. Figure 6 qualitatively shows the various regions of space and their associated values of the gap function, $g(\mathbf{x}^{(1)}, t)$, with a value of zero on $\gamma_c^{(2)}$. As a result, we may write the contact constraint as

$$g(\mathbf{x}^{(1)}, t) \leq 0. \tag{6.3}$$

The concept of the closest point projection to formulate the contact constraint was developed by T.A. Laursen and J.C. Simo [6] and has functioned as the historical foundation for the contact constraint expression in the continuous *and* discrete settings. While the pointwise constraint may be appropriate in the continuous setting, we will discuss its limitations in the numerical setting when used in node-to-segment methods, and face-on-face methods that are designed to overcome them.

[1] In the numerical contact literature, $\gamma_c^{(1)}$ would thus be referred to as the slave surface, while $\gamma_c^{(2)}$ the master surface

6.3 Strong Form of the Large Deformation Contact Problem

It remains to formulate the strong form of the initial boundary value problem for two deformable bodies, including contact interactions. This can be stated as follows:

$$
\begin{aligned}
&\text{find } \mathbf{u}^{(i)}(\mathbf{x}^{(i)}(\mathbf{X}^{(i)},t)), i=1,2, \text{such that,}\\
&\quad \nabla\cdot\mathbf{P}^{(i)}+\rho_0^{(i)}\mathbf{B}^{(i)}=\rho_0^{(i)}\mathbf{A}^{(i)}, \mathbf{X}\in\Omega_0^{(i)},\\
&\quad \mathbf{P}^{(i)}\mathbf{N}^{(i)}=\bar{\mathbf{p}}^{(i)}\in\Gamma_t^{(i)}\ \forall\ \ t\in[0,T], T>0,\\
&\quad \mathbf{u}^{(i)}=\bar{\mathbf{u}}\ \forall\ \mathbf{X}\in\Gamma_u^{(i)}, t\in[0,T], T>0,\\
&\quad \dot{\mathbf{u}}^{(i)}\rfloor_{t=0}=\bar{\mathbf{V}}^{(i)}\ \forall\ \mathbf{X}\ \ \in\bar{\Omega}_0^{(i)},\\
&\text{subject to the constraint,}\\
&\quad g(\mathbf{X}^{(1)},t)\le 0\ \forall\ \mathbf{X}^{(1)}\in\bar{\Omega}_0^{(1)},
\end{aligned}
\tag{6.4}
$$

where $\mathbf{P}^{(i)}$ is the first Piola Kirchoff stress associated with body i, whose constitutive model has been left unspecified for the sake of generality and simplicity.

The scalar valued gap function g is defined as,

$$
g(\mathbf{X}^{(1)},t)=-\mathbf{n}^{(2)}\cdot(\mathbf{x}^{(1)}-\bar{\mathbf{x}}_c^{(2)}),\ \forall\ \ \mathbf{x}_c^{(1)}\in\gamma_c^{(1)},
\tag{6.5}
$$

where $\mathbf{n}^{(2)}$ is the outward unit normal to $\gamma_c^{(2)}$ at the closest point $\bar{\mathbf{x}}_x^{(2)}$ defined in Equation 6.1. Note that the form of the scalar valued gap function pertains to the particular closest point projection formulation presented for the continuous two body contact case. Alternate forms of the kinmenatic constraint will be a principal consideration when examining contact in the numerical setting below.

7 Review of Numerical Methods in Contact

Computational methods in Lagrangian contact mechanics are primarily distinguished by the particular discretization of the weak form contact integrals, the expression of the kinematic constraint, and equilibrium enforcement across the contact interface. A central difficulty to all methods is determining how best to accomplish the aforementioned items in the presence of nonconforming, spatially discretized contact surfaces. The contact method presented in this work uses 2D and 3D linear finite elements whose surfaces are given by continuous, piecewise smooth surface approximations. The mathematical formulation is characterized by (1) discretization of the weak form contact integrals using a "median plane" methodology, (2) constructing a contact traction basis as a means of equilibrium enforcement, and (3) an integral expression for the kinematic constraint whose weak form is derived using the traction basis functions.

Some of the earliest computational contact methods are node-to-segment (NTS) methods, which were developed by Hallquist [7] and later used for self-contact by Benson [8]. In a NTS method, one surface is denoted as the slave surface, and the other the master. The contact interface is taken as the discretized master surface and *all* slave nodes are constrained to lie on this interface. This constraint is what often limits NTS methods in their ability to solve difficult contact problems; a difficulty that arises due to the fact that the kinematic constraint is enforced pointwise. The enforcement is typically over low-order surface approximations. When enforced pointwise, NTS methods experience jumps in the contact nodal forces during sliding due to discontinuous normals between adjacet surface facets. Furthermore, single-pass NTS methods that enforce unilateral constraints do not pass the contact patch test. Additionally, two-pass NTS methods that enforce bilateral constraints, which pass the patch test under certain conditions, often suffer from overconstraint, which manifests itself through mesh locking (see [9] for a thorough discussion on disadvantages

of node-to-segment methods). The interested reader may refer to [6] and [10] for in-depth treatements of NTS methods.

Severl authors have used surface smoothing techniques to ameliorate some of the issues associated with NTS methods. [11] provides higher order C^1 interpolation at the contact surfaces, while [12] introduces a smoothing gap algorithm based on a signed distance function between two contact surfaces aimed at addressing any lack of smooothness in NTS methods associated with discontinuous surface normals. Alternatively, [13] defines Gregory patches (refer to [14]) over each finite element surface facet in order to obtain a continuously differentiable surface normal. This method is shown to smooth contact force behavior in the presence of large sliding, a problem that plagues NTS methods, but is rather computationally expensive. An altogether different approach is pursued in [15] and [16], where a contact *domain* is defined to formulate kinematic quantities where each domain is principally and locally defined by a slave node and a master segment. This NTS-like method is shown to perform well for a variety of contact problems that would normally prove difficult for *traditional* NTS methods. The numerical examples, however, are largely limited to two dimensional analyses using constant strain triangular elements and a Neo-Hookian material model. Furthermore, mesh dependencies are introduced by retaining the master-slave notion under which the contact domains are defined, and by introducing a user prescribed stability parameter, which allows the user to avoid overconstraint by relaxing the degree to which the pointwise kinematic constraint is enforced. The most modern computational contact methods, however, represent a departure from NTS formulations and consider the contact constraints and enforcement over face-pair overlaps between contact surfaces.

The next evolution in contact algorithms was influenced by mortar methods used in domain decomposition (see [17]). Early work on these methods can be found in [18], while a significant mortar-based contact method was developed for large deformation solid mechanics in [9] by Puso and Laursen. This work was extended to include friction [19] and use with

quadratic elements [20]. Subsequent efforts (e.g. [21–23]) have established mortar-based contact methods as the most recent and successful advance in computational approaches to contact. These methods identify non-mortar and mortar surfaces (analogous to master/slave surfaces in to node-to-segment lexicon), and often use unilateral constraints, making them single-pass like methods. In [9], the authors define a *median plane* onto which two contacting facets are projected, whose intersection (overlap) defines an intermediate surface over which weak form contact integrals are evaluated. In that work, the median plane is generally taken as the plane in which the non-mortar face lies, with special treatment for warped non-mortar faces. A piecewise-constant traction is defined over each intermediate surface, whose distribution is defined over the collection of all piecewise-planar, but discontinuous, face-pair overlaps. Additionally, an integral form of the kinematic constraint is used, which smoothes contact behavior in the presence of sliding, and alleviates the issue of locking that plagues NTS methods for certain types of contact problems. Like all mortar methods, the unknown Lagrange multiplier field and the kinematic constraint are interpolated on the non-mortar side.

The mortar-based methods still introduce non-mortar/mortar surface designations where biased formulations interpolate the kinematic constraint and unknown contact tractions using the finite element interpolants on the non-mortar side. A mesh dependency is also introduced when the non-mortar surface is used to discretize the contact interface, which introduces a sort of asymmetry in the contact formulation. If an unbiased two-pass like method were used to address this, then twice as many traction unknowns are introduced, thereby increasing computational cost.

This work presents an unbiased face-on-face contact formulation for 3D, large deformation quasi-static solid mechanics with material nonlinearity outside of the traditional mortar-method framework. In this work symmetric median planes are defined, local to each face-on-face interaction, which in turn are used define intermediate surfaces that serve to discretize

the contact interface. The unknown contact traction vectors defined on each intermediate surface are decomposed into piecewise constant tangential vectors and a normal component defined by a piecewise linear pressure. Rather than interpolating the traction field on one side of the contact interface, which results in a biased method, or interpolating the traction field on both sides, which introduces twice as many unknowns, this work abandons the non-mortar/mortar parameterization and introduces an "up to" linear pressure basis on the symmetric intermediate surfaces at contacting face-pairs. Not only is the linear pressure basis used to define the pressures themselves, but the basis functions are used as test functions in the weak form of the kinematic constraints. Some unique features of this traction basis methodology is that equilibrium across the contact interface is guaranteed, sparsity of the contact contributions to the system of equations is preserved, the formulation allows for an enriched pressure basis, and the active set of contacting face-pairs is successfully controlled through a unique subcycle procedure. Prior to the presentation of the mathematical formulation, various face-face interface discretizations are reviewed.

7.1 A Survey of Face-on-Face Contact Interface Discretizations

A diagramatic representation of contact between two finite element surfaces using piecewise linear surface approximations is shown in Figure 7. While contact in the continuous setting may be resolved over a single interface γ_c, contact in the discretized setting must handle the fact that $\gamma_c^{(1)} \cap \gamma_c^{(2)} = \{p_a^k\}$, where $\{p_a^k\}$ denotes the set of possibly non-contiguous polytopal intersections of degree less than or equal to k. In general, $\gamma_c^{(1)}$ and $\gamma_c^{(2)}$ will be noncoincident and nonconforming, where for 3D contact $k \leq 2$ and for 2D contact $k \leq 1$. The fact that contact interactions have to be resolved over non-conforming, noncoincident piecewise polytopal interfaces provides strong motivation to devise a disretization that defines a single γ_c over which to numerically resolve contact. Doing so, in combination with an integral form of the contact kinematic constraint, is an advantageous departure from NTS methods. However, as will be highlighted, many discretization methods introduce an asymmetry favoring one contact surface mesh over the other.

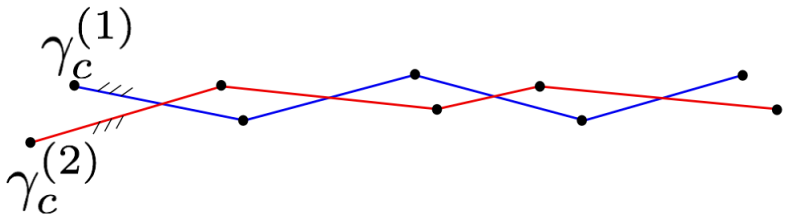

Figure 7: A 2D contact interaction between two representative portions of the contact surfaces $\gamma_c^{(1)}$ and $\gamma_c^{(2)}$, which are composed of 1D segments.

Mortar methods handle the contact surfaces by designating mortar and non-mortar sides of the interface. The discretization of the contact interface and the interpolation of kinetic and kinematic quantities is done using the interpolants on the non-mortar side. This single surface parameterization introduces unilateral constraints and results in a biased, single-pass like method. One may, however, adopt a mortar-*like* method that discretizes the contact

interface in another manner. The following subsections introduce various ways in which this discretization can be handled. There is a slave-to-master projection, which is mortar-*like*; the typical non-mortar projection used in mortar methods, and an altogether separate median-plane projection.

7.1.1 Slave-to-Master Projection

One spatial discretization of the contact interface is derived from identifying the portion of a given non-mortar segment that interpenetrates its associated mortar segment. That is, we look at contact locally in unique mortar/non-mortar pairs. This method is based on a discussion with Nate Crane of Sandia National Laboratories on contact implementations in their solid mechanics code. This local interaction, shown in Figure 8, forms the foundation for the discretization of γ_c.

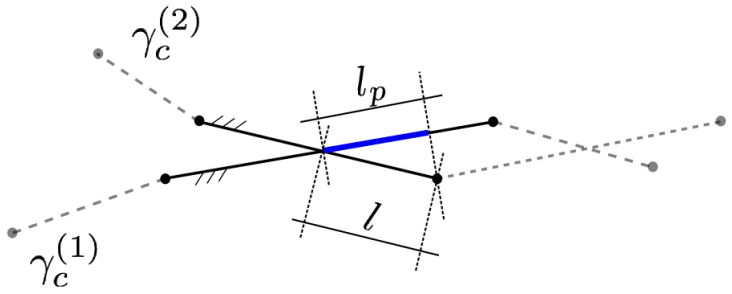

Figure 8: A mortar segment is shown on surface $\gamma_c^{(1)}$ and a non-mortar segment on surface $\gamma_c^{(2)}$. The kinematically inadmissable portion of the non-mortar segment, l, is orthogonally projected onto the mortar segment. The projection is denoted l_p.

The orthogonal projection of the violating portion of the non-mortar segment, l, onto the mortar segment is shown in Figure 8 and is denoted as l_p. This length forms a local segment-segment constituent of γ_c over which local contact contributions to the residual are integrated. This projection method may be a logical choice when viewed from the perspective

that the non-mortar segment is kinematically constrained to the mortar segment. Using an integral form for the kinematic constraint, this means that the non-mortar segment must have zero mean gap over its projected length (or area) onto the mortar segment.

7.1.2 Image Plane Projections

The slave-to-master projection presented in the previous section is best understood when looking at a face-on-face interaction edge on, as in Figure 8. This is because one must actually identify to what extent a slave face interpenetrates its associated master face in order to form the local contact intermediate surface. Another way to view a contact interaction is from above, or from a "bird's eye" view. This is a method, herein referred to as "image plane projection," that is used in [9]. Figure 9 shows a "bird's eye" view of two facets that, for the purposes of this discussion, are assumed to be in contact. The two facets are seen to overlap, forming the intermediate surface (or region) denoted as ω_α. Each face-face interaction forms a unique ω_α. The union of all such surfaces form γ_c. While the precise definition of the image plane is a matter of interpretation, this is the discretization method used in [9] and is the method adopted in this work.

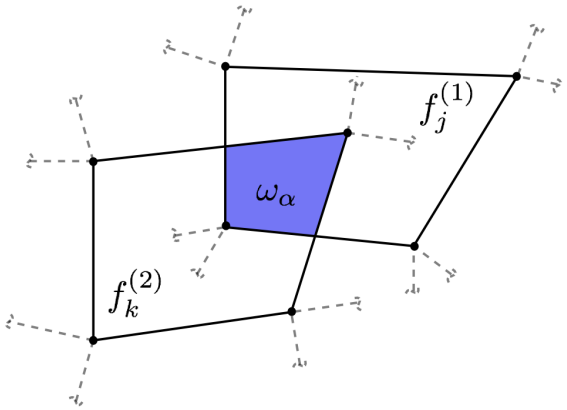

Figure 9: Face $f_j^{(1)}$ is shown to overlap face $f_k^{(2)}$ where $f_j^{(1)} \cap f_k^{(2)} = \omega_\alpha$, the local region of overlap defining the intermediate surface.

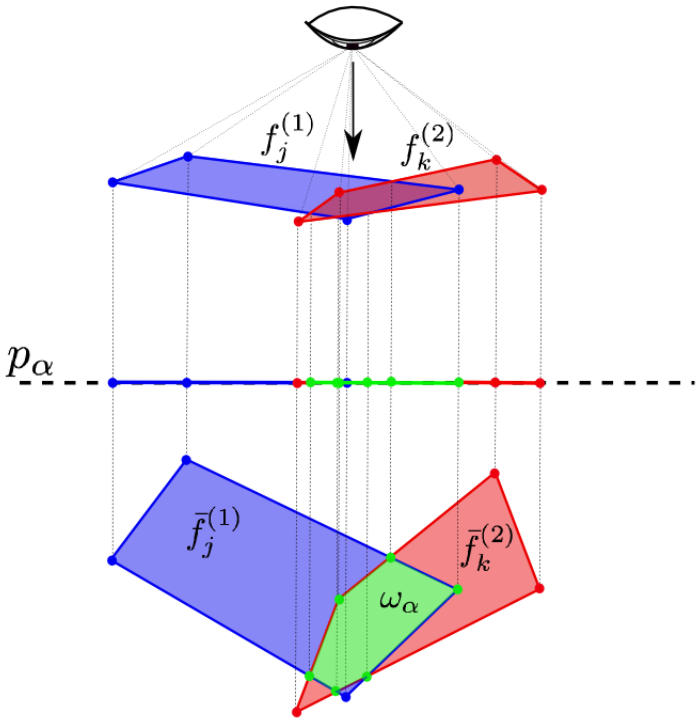

Figure 10: Faces $f_j^{(1)}$ and $f_k^{(2)}$ are viewed from above, perpendicular to the image plane, p_α. The observer sees the face images, $\bar{f}_j^{(1)}$ and $\bar{f}_k^{(2)}$, which are the projections of $f_j^{(1)}$ and $f_k^{(2)}$ onto p_α.

The face-face interaction shown in Figure 9 was loosely said to be viewed "from above," when in fact the overlap (i.e. intermediate surface) ω_α depends on the observer's viewing direction. That is, the overlap is constructed in the observer's eye as the projection of the faces onto the observer's image plane, which is the plane perpendicular to the observer's viewing direction. This concept is illustrated in Figure 10.

In the ideal case, $f_j^{(1)}$ and $f_k^{(2)}$ are coplanar in a face-face interaction. The observer's image plane will then be the plane of the two faces, which will overlap to form a coplanar region ω_α. In general, however, this will not be the case. The most general case occurs when face $f_j^{(1)}$ is not coplanar with face $f_k^{(2)}$. As a result, one must make a decision as to the image plane chosen with which to define the face-face intersection. This is the central idea behind a projection of each face onto a common, or *median*, plane where the resulting overlapping region, ω_α, is the intermediate surface that represents a local contact interface over which contact integrals are evaluated.

To understand the effect of this local discretization method, consider the following example. Figure 11 shows two cubes in two contact interactions, as well as the associated contact interfaces, shaded in gray. The top cube has twice the mesh density as the bottom cube. The interaction on the left shows a coincident, but nonconforming interaction in which the two cubes are aligned. The interaction on the right shows a coincident (i.e. coplanar), but nonconforming interaction where the top cube is rotated 45 degrees. In each case, both sides of the contact interface are coplanar, so the image plane is one and the same as the contact surfaces. Figure 12 shows a plan view of the contact interfaces associated with each contact configuration shown in Figure 11, and the resulting intermediate surfaces. In particular, this figure shows each ω_α region associated with each local face-face interaction. Collectively, these ω_α regions constitute the discretization of the contact interface.

One such image plane projection is the non-mortar projection used in [9]. Mortar methods

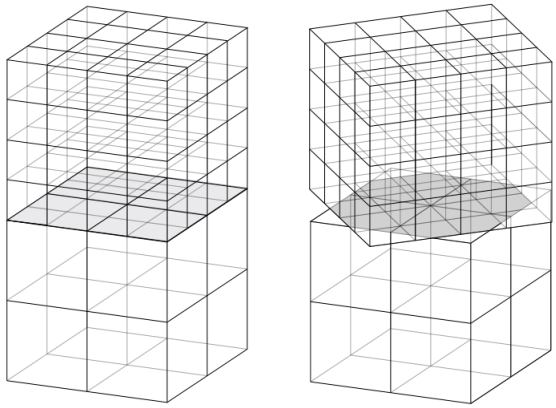

Figure 11: Two different contact interactions between two cubes. The top has twice the mesh density as the bottom.

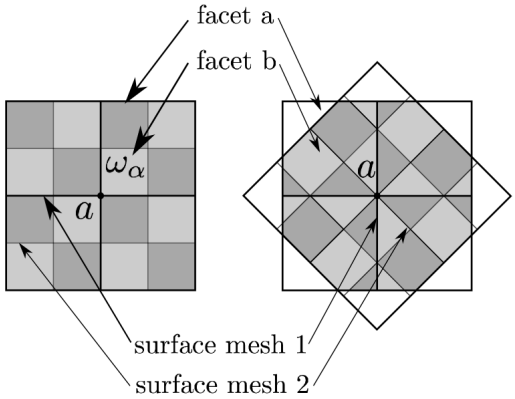

Figure 12: The left and right interfaces correspond to the cube-cube interactions shown in Figure 11. ω_α overlaps are shown in gray. The center node a belonging to the coarser mesh is shown.

interpolate kinetic and kinematic quantities on the non-mortar surface. As a result, it is a logical choice to align one's viewing direction with the outward facing normal of the non-mortar face. The non-mortar face then forms the observer's image plane, and therefore a local median plane with intermediate surface patches, ω_α.

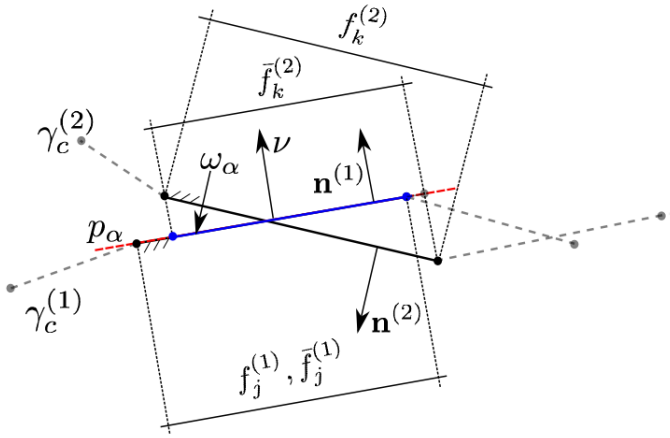

Figure 13: Mortar face $f_k^{(2)}$ belonging to $\gamma_c^{(2)}$ is projected onto non-mortar face $f_j^{(1)}$ forming a projected face image, $\bar{f}_k^{(2)}$, on the median plane, p_α. The intersection, $\omega_\alpha = \bar{f}_j^{(1)} \cap \bar{f}_k^{(2)}$, forms the local intermediate contact surface.

Figure 13 shows an edge on view of an interaction between the mortar face $f_k^{(2)}$ and the non-mortar face $f_j^{(1)}$. The mortar facet is projected onto the non-mortar facet, forming a projected image. The mortar and non-mortar facet images, $\bar{f}_k^{(2)}$ and $\bar{f}_j^{(1)}$, respectively, are used to construct a local region of overlap, ω_α, shown by the blue line, which lies in the image plane p_α. The unit normal to each facet is denoted as $\mathbf{n}^{(i)}, i = 1, 2$. In this case, the unit normal to the median plane, $\boldsymbol{\nu}$ is in fact $\mathbf{n}^{(1)}$. This method of discretizing γ_c is used in [9].

8 An Image Plane Variant for the Discretization of the Contact Interface

A variant of the median-plane projection method used in [9] is introduced and used in this work. This variant discretizes the contact interface in a way that is not biased toward one surface or the other, which falls outside the mortar framework. Instead, local intermediate surfaces are formed that do not lie on either of the actual contacting surfaces. The intermediate surfaces are fictitious surfaces that are formed in a way that is symmetric with respect to each of the contacting faces in a given, local face-pair. The discretization method used in this work is described below.

First, the contact discretization is defined by local face-face interactions. For 3D linear finite elements, this results in local interactions between four node quadrilateral facets, which may be warped. For 2D linear finite elements, this results in local interactions between two node line segements. A given face may be refered to as a facet or segment as dimensionally appropriate, but in all cases will be denoted as $f_j^{(i)}$, which is the j^{th} face on the i^{th} contact surface. This work considers two body flexible contact with $i = 1, 2$ and $j \leq \text{ncf}^1$ for $i = 1$ and $j \leq \text{ncf}^2$ for $i = 2$, where ncf^1 and ncf^2 are the number of contacting faces on surfaces 1 and 2, respectively. Figure 14 shows two quadrilateral facets in contact, as viewed from above. From this perspective, we observe the facets as projected onto an image plane, or a plane perpendicular to the observer's line of sight. Figure 14 shows $\omega_\alpha = f_j^{(1)} \cap f_k^{(2)}$ as the facet-facet *intersection* show in blue. If an observer's line of sight were to align perpendicularly to one of the facet planes, then this will serve as the image plane, defining both facet projections and their intersection. This work, however, seeks an unbiased, symmetric discretization and formulation. As a result, we define an image plane as a geometrical entity informed by, but altogether separate from the two faces that form a local contact interaction.

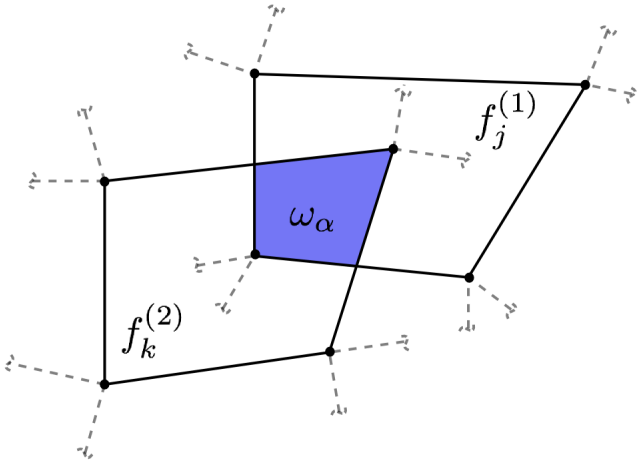

Figure 14: Facet $f_j^{(1)}$ is shown to overlap facet $f_k^{(2)}$ where $f_j^{(1)} \cap f_k^{(2)} = \omega_\alpha$, the local region of overlap or intermediate contact surface.

Figure 15 shows the proposed median plane in relation to each facet from which it is constructed. The edge-on view shows the local face-face contact interaction between facets $f_j^{(1)}$ and $f_k^{(2)}$ and the median plane, p_α. One way to define the median plane is to derive its unit normal by *averaging* the current configuration outward unit normals, $\mathbf{n}^{(1)}$ and $\mathbf{n}^{(2)}$, belonging to $f_j^{(1)}$ and $f_k^{(2)}$. With this approach, equation.

$$\boldsymbol{\nu} = \frac{1}{2}\frac{\mathbf{n}^{(1)} - \mathbf{n}^{(2)}}{\|\mathbf{n}^{(1)} - \mathbf{n}^{(2)}\|}. \tag{8.1}$$

This work proposes an alternative method, which is presented later on with the concept of the active set in Section 9.4.2. Problems, however, were run using Equation 8.1 and this definition of the median plane was seen to be perfectly adequate.

Some notation regarding material point projections must be laid out to facilitate the development of concepts contained in this work. Specifically, $\bar{f}_j^{(1)}$ and $\bar{f}_k^{(2)}$ are formed by projecting the nodes of $f_j^{(1)}$ and $f_k^{(2)}$ onto p_α. Let us define a projection operator, $\mathcal{P}_\alpha^{(i)} : \mathbf{x}^{(i)} \to \bar{\mathbf{x}} \,\forall\, \mathbf{x}^{(i)} \in f_r^{(i)}$, where $r = \{j, k\}$ for $i = 1, 2$, and $\bar{\mathbf{x}} \in \omega_\alpha = \bar{f}_j^{(1)} \cap \bar{f}_k^{(2)}$. Conversely, material points on

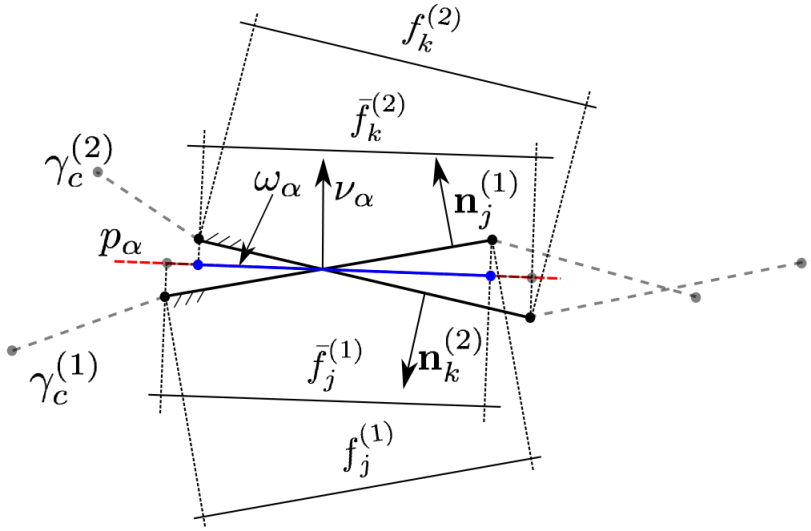

Figure 15: The image plane, p_α, is formed from the calculated normal vector, ν, and represents the median plane between facets $f_j^{(1)}$ and $f_k^{(2)}$. The facet images, $\bar{f}_j^{(1)}$ and $\bar{f}_k^{(2)}$ are projections of $f_j^{(1)}$ and $f_k^{(2)}$ onto p_α, respectively. The region, $\omega_\alpha = \bar{f}_j^{(1)} \cap \bar{f}_k^{(2)}$ is the local intermediate contact surface.

each ω_α are projected back to p_α's respective $f_j^{(1)}$ and $f_k^{(2)}$ pair in the direction of $\boldsymbol{\nu}$ via the projection operator $\bar{\mathcal{P}}_\alpha^{(i)} : \bar{\mathbf{x}} \rightarrow \mathbf{x}^{(i)} \; \forall \; \mathbf{x}^{(i)} \in f_r^{(i)}, r = \{j, k\}, i = 1, 2$ and $\bar{\mathbf{x}} \in \omega_\alpha$. Furthermore, let $\omega_\alpha^{(i)}$ denote the projection of ω_α in the direction of $\boldsymbol{\nu}$ onto surface i where it is understood that the projection is either onto $f_j^{(1)}$ or $f_k^{(2)}$ that were used to form ω_α in the first place.

In reference to Figure 15, the exact point location of the plane does not factor into the formulation, but p_α is shown to cut through the local interaction in a symmetric fashion. The point through which it is shown to pass is computed by averaging the centroids of the polytopes that result from projecting ω_α back onto faces $f_j^{(1)}$ and $f_k^{(2)}$; that is, the centroids of $\omega_\alpha^{(1)}$ and $\omega_\alpha^{(2)}$. Additionally, in the case of warped quadrilateral facets, an average facet normal located at the vertex averaged centroid of the facet is calculated using Stokes' Theorem and used in Equation 8.1.

The intermediate surface is formed by the orthogonal projection of faces $f_j^{(1)}$ and $f_k^{(2)}$ onto p_α, which form the facet images, $\bar{f}_j^{(1)}$ and $\bar{f}_k^{(2)}$, on the median plane. The intersection of

$\bar{f}_j^{(1)}$ and $\bar{f}_k^{(2)}$ forms a local overlap and defines the intermediate surface, ω_α. This overlap solely characterizes a face-face contact interaction and serves as a local constituent in the discretization of the contact interface and weak form contact integrals. Lastly, as a parenthetical remark regarding the interface geometry and the contact boundaries, $\gamma_c^{(1)}$ and $\gamma_c^{(2)}$, it is worth discussing some geometrical aspects of the median plane projection.

Figure 16 shows a face-face interaction. The two faces belonging to $\gamma_c^{(1)}$ are denoted $f_j^{(1)}$ and $f_{j+1}^{(1)}$, whereas the single red face belonging to $\gamma_c^{(2)}$, is denoted $f_k^{(2)}$. Each face on $\gamma_c^{(1)}$ is shown in blue and interacts with the face on $\gamma_c^{(2)}$, shown in red. The subsequent median planes, p_α and $p_{\alpha+1}$, and the green regions of overlap ω_α and $\omega_{\alpha+1}$, are also shown. What is different in this figure is that the regions of overlap are then projected *back* onto $f_j^{(1)}$, $f_{j+1}^{(1)}$, and $f_k^{(2)}$ in the direction normal to each of their respective median planes. These images on $f_j^{(1)}$ and $f_{j+1}^{(1)}$ are denoted $\omega_\alpha^{(1)}$ and $\omega_{\alpha+1}^{(1)}$ whereas the images on $f_k^{(2)}$ are denoted $\omega_\alpha^{(2)}$ and $\omega_{\alpha+1}^{(2)}$. The point to be made here is that the intersection, $\omega_\alpha^{(2)} \cap \omega_{\alpha+1}^{(2)} \neq \emptyset$. Another observation is made by considering the face-face interaction shown in Figure 17. The notation and colored regions have the same meaning as in Figure 16, but in this case the $\omega_\alpha^{(i)}$ and $\omega_{\alpha+1}^{(i)}$, i = 1,2, regions do not intersect. Rather, the regions $l^{(1)}$ and $l^{(2)}$ are regions on $f_j^{(1)}$ and $f_{k+1}^{(2)}$ that are not covered by the projection of ω_α and $\omega_{\alpha+1}$ back onto their respective associated faces.

There is no inherent significance in the geometrical observations being made regarding the image of a region of overlap on the respective faces that are used to form that overlap. Rather, these notes are simply used to express observations of the characteristics of the median plane projection. The projection of each region of overlap, ω_α, back onto its associated faces, $f_j^{(1)}$ and $f_k^{(2)}$, will be discussed in some detail for the purposes of numerical quadrature. As a result, the development of this concept and corresponding notation is important. As a final remark, while we have expressed γ_c as $\gamma_c = \cup_{\alpha=1}^{N} \omega_\alpha$, it is actually the case that $\gamma_c^{(i)} \neq \cup_{\alpha=1}^{N} \omega_\alpha^{(i)}$, i = 1,2, for reasons that can be understood by close examination of Figures 16 and 17. There is nothing that states this ought not be the case. In fact, we are not concerned

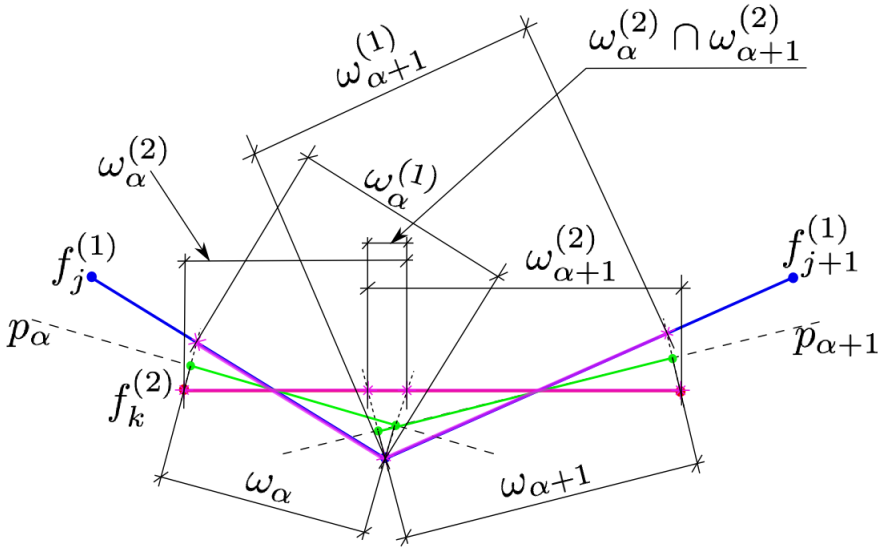

Figure 16: An example face-face interaction is shown edge-on. The blue faces are on $\gamma_c^{(1)}$, while the single red face is on $\gamma_c^{(2)}$. The green are the respective ω_α and $\omega_{\alpha+1}$ overlap regions. The purple regions are the projections of the green overlaps back onto each facet. Notice that the purple regions projected onto the red facet actually overlap.

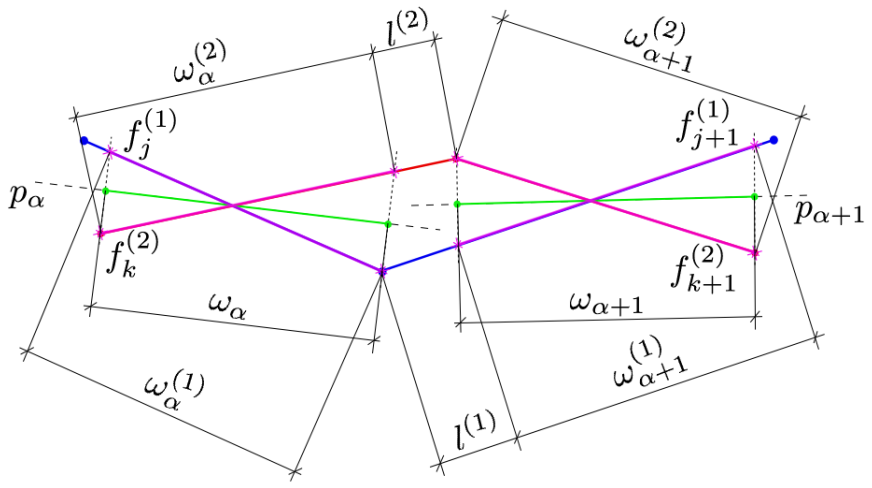

Figure 17: An example face-face interaction is shown edge-on. The blue facet belong to $\gamma_c^{(1)}$, while the red faces belong to $\gamma_c^{(2)}$. The green denote the respective overlap regions, ω_α and $\omega_{\alpha+1}$. The purple are the projections of the green regions back onto each facet. The regions $l^{(1)}$ and $l^{(2)}$ denote the regions on $\gamma_c^{(1)}$ and $\gamma_c^{(2)}$ that are not covered by this projection.

with the geometric interaction of a ω_α and $\omega_{\alpha+1}$ as our discretization of γ_c is entirely local at a *single* face-face interaction level.

9 Mathematical Formulation of a New Face-on-Face Contact Methodology

The new face-on-face contact methodology presented herein is developed for implicit analysis with finite deformations and material nonlinearity. The weak form of the inital boundary value problem presented in Equation 4.18, and the finite element residual presented in Equation 4.34, are simplified in that the acceleration terms are dropped and time integration is not required. As a result, the following presentation explicitly considers quasi-static analysis, but the concepts pertaining to the contact formulation hold for dynamic analysis without the aforementioned simplifications to the weak form and the finite element residual.

The formulation presented in this section is based on the continuum description of large deformation contact, which is outlined in Figure 5. The following section will introduce the weak form and finite element residual with the aforementioned simplifications appropriate to quasi-static analysis.

9.1 Weak Form of the Equations of Motion and the Finite Element Residual

Let us define the solution space, $\mathcal{S}^{(i)}$, and weighting space, $\mathcal{V}^{(i)}$ defined on the domain $\bar{\Omega}_0^{(i)}$

$$\mathcal{S}^{(i)} = \{\mathbf{u} \mid \mathbf{u} \in H^1(\Omega^{(i)}),\ \mathbf{u} = \bar{\mathbf{u}} \text{ on } \Gamma_u^{(i)}\} \tag{9.1}$$

and

$$\mathcal{V}^{(i)} = \{\mathbf{v} \mid \mathbf{v} \in H^1(\Omega^{(i)}),\ \mathbf{v} = \mathbf{0} \text{ on } \Gamma_u^{(i)}\}, \tag{9.2}$$

where $i = 1, 2$ for each of the two bodies in contact, $\Gamma_u^{(i)}$ denotes the boundary over which Dirichlet boundary conditions are specified and H^1 is the usual Sobolev space of order one.

The weak form of the quasi-static equations of motion for the two flexible body contact problem in finite deformations is as follows: find $\mathbf{u}^{(i)} \in \mathcal{S}^{(i)}$ such that,

$$\sum_{\imath=1}^{2} \left(\int_{\Omega_0^{(i)}} P_{jk}^{(i)} v_{j,k}^{(i)} dV - \int_{\Omega_0^{(i)}} \rho_0^{(\imath)} b_j^{(\alpha)} v_j^{(i)} dV - \int_{\Gamma_t^{(i)}} \bar{p}_j^{(i)} v_j^{(i)} dA - \int_{\Gamma_c^{(i)}} \hat{p}_j^{(i)} v_j^{(i)} dA \right) = 0, \quad (9.3)$$

$\forall\ \mathbf{v} \in \mathcal{V}^{(i)}, i = 1, 2.$

$\mathbf{P}$ is the first Piola Kirchhoff stress tensor, $\mathbf{b}$ is a prescribed body force per unit mass, $\bar{\mathbf{p}}$ is the given Piola traction, and $\hat{\mathbf{p}}$ is the Piola contact traction on the contact portion of the boundary. Note that we have written $P_{jk}^{(i)}$ rather than $P_{j\beta}^{(i)}$ as was done in the earlier sections of this work where the Greek subscript was used to denote the reference configuration. This was done in the earlier sections for clarity in introducing different stress measures. Heretofore, however, Greek subscripts, namely α, are reserved for contact related entities associated with the discretization of the contact interface. It is clearer to introduce Greek subscripts for contact related entities than confuse them with the standard i, j, k for vectors and tensors for components related to physical space.

Let us proceed with the standard Galerkin approximation to the weak form with the following approximation of the test functions $\mathbf{v}$ in Equation 9.3,

$$\mathbf{v}^h(\mathbf{X}) = \sum_{a=1}^{N} \mathbf{v}_a \varphi_a(\mathbf{X}), \quad (9.4)$$

where $v^h \in \mathcal{V}^h \subset \mathcal{V}$, φ is a basis for $\mathcal{V}^h$, and N is the total number of nodes in the discrete problem. $\mathcal{V}^h$ is the typical finite dimensional subspace associated with the discretized finite element problem. The discrete finite element residual at node a is formed by substituting

Equation 9.4 into Equation 9.3 leading to the following expression:

$$R_{ja} = \int_{\Omega_0^{(i)}} P_{jk}^{(i)} \varphi_{a,k}^{(i)} dV - \int_{\Omega_0^{(i)}} \rho_0^{(i)} b_j^{(i)} \varphi_a^{(i)} dV - \int_{\Gamma_t^{(i)}} \bar{p}_j^{(i)} \varphi_a^{(i)} dA - \int_{\Gamma_c^{(i)}} \hat{p}_j^{(i)} \varphi_a^{(i)} dA = 0. \quad (9.5)$$

Equation 9.5 may be written as follows,

$$R_{ja} = \bar{R}_{ja} + \hat{R}_{ja} = 0, \quad (9.6)$$

where

$$\bar{R}_{ja} = \int_{\Omega_0^{(i)}} P_{jk}^{(i)} \varphi_{a,k}^{(i)} dV - \int_{\Omega_0^{(i)}} \rho_0^{(i)} b_j^{(i)} \varphi_a^{(i)} dV - \int_{\Gamma_t^{(i)}} \bar{p}_j^{(i)} \varphi_a^{(i)} dA, \quad (9.7)$$

denotes the portion of the residual arising from the non-contact boundary value problem and

$$\hat{R}_{ja} = - \int_{\Gamma_c^{(i)}} \hat{p}_j^{(i)} \varphi_a^{(i)} dA, \quad (9.8)$$

denotes the additional contact portion of the residual. The emphasis in this work is on the nature and formulation of the traction, $\hat{\mathbf{p}}^{(i)}$ in Equation 9.8.

9.2 Contact Traction Formulation

The formulation of the contact traction used in the integral term of Equation 9.8 is designed to guarantee equilibrium across the contact interface.

9.2.1 Contact Equilibrium and Interface Discretization

It is useful to express the contact integrals in Equation 9.5 in the current configuration since the current configuration is where contact will be enforced, and is most intuitively understood to occur. Let $\hat{\mathbf{f}}_a^{(i)}$ denote the contact force contribution to the residual at node

a on surface i. We can express the two contact nodal force contributions as follows:

$$\hat{f}_{ia}^{(1)} = \int_{\gamma_c^{(1)}} \hat{t}_i^{(1)} \varphi_a^{(1)} da, \tag{9.9}$$

and

$$\hat{f}_{ib}^{(2)} = \int_{\gamma_c^{(2)}} \hat{t}_i^{(2)} \varphi_b^{(2)} da, \tag{9.10}$$

where $\hat{\mathbf{t}}^{(i)}$ is the Cauchy contact traction on surface i. To be clear, a refers to the nodes on side (1) and b refers to the nodes on side (2). The previous equations are written in a way that implies $a \neq b$, where each side has a unique node numbering. Equilibrium across the contact interface is achieved when $\hat{\mathbf{t}}^{(1)} = -\hat{\mathbf{t}}^{(2)}$. As a result, we proceed by denoting a single contact traction distribution, $\hat{\mathbf{t}}$, and rewrite Equations 9.9 and 9.10 as follows.

$$\hat{f}_{ia}^{(1)} = \int_{\gamma_c^{(1)}} \hat{t}_i \varphi_a^{(1)} da, \tag{9.11}$$

and

$$\hat{f}_{ib}^{(2)} = -\int_{\gamma_c^{(2)}} \hat{t}_i \varphi_b^{(2)} da. \tag{9.12}$$

Contact equilibrium may then be stated as,

$$\sum_a \left(\int_{\gamma_c^{(1)}} \hat{\mathbf{t}} \varphi_a^{(1)} da - \int_{\gamma_c^{(2)}} \hat{\mathbf{t}} \varphi_a^{(2)} da \right) = \mathbf{0}, \tag{9.13}$$

where it is understood that $\varphi_a^{(i)} = 0$ for a not belonging to side i. and the nodal interpolation is now occuring based on the current configuration. As a final note regarding the nodal subscript, a, it is convenient to express Equation 9.13 as a single sum over all contact nodes. This means that a must range over the nodes on side (1) *and* side (2). This is a simplification and departure from the subscripting used in Equations 9.9 and 9.10, where independent node numbering was a more logical choice. For Equation 9.13 to hold, the area of both $\gamma_c^{(1)}$ and

$\gamma_c^{(2)}$ over which $\hat{\mathbf{t}}$ acts must be the same; otherwise, the contact tractions may be equal-and-opposite, but the sum of contact nodal forces on each side of the contact interface will not be. As a result, recall the method of discretizing the contact interface used in this work presented in Section 8. We use the median plane projection method outlined in Section 7.1.2 to compose a single contact interface, γ_c, consisting of unique, but disjoint, overlapping regions, $\omega_\alpha, \alpha = 1, 2, ..., N$ that are formed from local face-face interactions. As a result, we take $\gamma_c^{(i)} \approx \gamma_c = \cup_\alpha \omega_\alpha$. Contact equilibrium may then be expressed as

$$\sum_a \sum_\alpha \left(\int_{\omega_\alpha} \hat{\mathbf{t}}_\alpha \bar{N}_a^{(1)} da - \int_{\omega_\alpha} \hat{\mathbf{t}}_\alpha \bar{N}_a^{(2)} da \right) = \mathbf{0}, \tag{9.14}$$

where $\hat{\mathbf{t}}(\bar{\mathbf{x}}) = \hat{\mathbf{t}}_\alpha \; \forall \; \bar{\mathbf{x}} \; \in \omega_\alpha$ is the contact traction distribution defined specifically on each ω_α. Most importantly, $\bar{N}_a^{(i)}$ is the finite element shape function associated with node a defined over the face *image*, $\bar{f}_j^{(i)}$, which is the projection of $f_j^{(i)}$ onto p_α (note that faces $\bar{f}_j^{(1)}$ and $\bar{f}_k^{(2)}$ whose intersection forms an ω_α also each contain the region, ω_α). That is, a local contact interaction that occurs between faces $f_j^{(1)}$ and $f_k^{(2)}$ on $\gamma_c^{(1)}$ and $\gamma_c^{(2)}$, respectively, is being *resolved* over $\bar{f}_j^{(1)}$ and $\bar{f}_k^{(2)}$ on the median plane, p_α. As a result, a basis function over γ_c does not exist since γ_c is in fact a collection of disjoint ω_α. Rather, the integral terms in Equation 9.14 contain finite element shape functions associated with each of the projected faces that form an intermediate surface, ω_α. Specifically, a given projected facet admits the isoparametric mapping,

$$\bar{\mathbf{x}}(\boldsymbol{\xi}) = \sum_a N_a(\xi) \bar{\mathbf{x}}_a, \tag{9.15}$$

where $\boldsymbol{\xi}$ is the vector defining the parent space of the projected finite element face, and $\bar{\mathbf{x}}_a$ are the projected face's nodal coordinates. The shape functions in Equation 9.14 are taken as functions of $\bar{\mathbf{x}} \in \bar{f}_j^{(i)}$ on p_α by virtue of the integral over ω_α. That is, $N_a = N_a(\boldsymbol{\xi}(\bar{\mathbf{x}}))$, whose evaluation at a given $\boldsymbol{\xi}$ is determined using an inverse isoparametric mapping of $\bar{\mathbf{x}} \rightarrow \boldsymbol{\xi}$ (the inverse isoparametric mapping is discussed in Section 9.5). For clarity, a bar has been added over N_a in Equation 9.14 to clarify that $\bar{N}_a = N_a(\boldsymbol{\xi}(\bar{\mathbf{x}}))$ is a function of position on the

intermediate surface. Formulated in this manner, Equation 9.14 not only enforces an equal-and-opposite contact traction, but also ensures equal-and-opposite nodal forces through the use of discrete, but common, intermediate surfaces at the local face-face interaction level.

9.2.2 Contact Traction Distribution

The contact interface, γ_c, is formed by the union of the disjoint intermediate surfaces, ω_α, formed from local face-face interactionss. As a result, it makes sense to describe the full contact traction distribution as the union of local traction distributions on each intermediate surface, as is expressed in Equation 9.14. The local traction distribution is decomposed into tangential and normal components as follows,

$$\hat{\mathbf{t}}_\alpha(\bar{\mathbf{x}}) = \hat{\mathbf{t}}_\alpha^{(n)} + \hat{\mathbf{t}}_\alpha^{(t)} \quad \forall \ \bar{\mathbf{x}} \in \omega_\alpha, \tag{9.16}$$

where

$$\hat{\mathbf{t}}_\alpha^{(n)}(\bar{\mathbf{x}}) = \hat{p}_\alpha \bar{\mathbf{n}}_\alpha \quad \forall \ \bar{\mathbf{x}} \in \omega_\alpha, \tag{9.17}$$

is the normal traction component and

$$\hat{\mathbf{t}}_\alpha^{(t)}(\bar{\mathbf{x}}) = \hat{\boldsymbol{\tau}}_\alpha \quad \forall \ \bar{\mathbf{x}} \in \omega_\alpha, \tag{9.18}$$

is the tangential traction component, where $\hat{\boldsymbol{\tau}}_\alpha \cdot \bar{\mathbf{n}}_\alpha = 0$. Each overlap, ω_α, has an associated contact pressure, $\hat{p}_\alpha$, and unit normal, $\bar{\mathbf{n}}_\alpha$, defining the median plane, p_α. Additionally, each overlap has an associated shear traction, $\hat{\boldsymbol{\tau}}_\alpha$. The contact pressure p_α, is taken to be piecewise linear over each ω_α region, while the shear traction, $\hat{\boldsymbol{\tau}}_\alpha$ is taken to be piecewise constant over each ω_α. For the purposes of enforcing *normal* contact and no interpenetration of $\Omega^{(1)}$ and $\Omega^{(2)}$, we will focus on the pressure term in Equation 9.16 only. Note that $\hat{\boldsymbol{\tau}}$ is nonzero in the presence of frictional contact, but does not play a role in the enforcement of

the kinematic constraint.

Let us proceed with a full three-dimensional treatment of contact and represent $\hat{p}_\alpha$ as an arbitrary linear polynomial as follows:

$$\hat{p}_\alpha = q_0^\alpha + r_1^\alpha x + r_2^\alpha y, \quad \forall \; x, y \in \omega_\alpha, \tag{9.19}$$

where (x, y) are spatial coordinates with respect to a coordinate system local to the median plane, p_α. Plugging Equation 9.19 into Equation 9.17 gives

$$\hat{\mathbf{t}}_\alpha^{(n)} = (q_0^\alpha + r_1^\alpha x + r_2^\alpha y)\bar{\mathbf{n}}_\alpha, \quad \forall \; x, y \in \omega_\alpha. \tag{9.20}$$

Equation 9.11 and 9.12 can be rewritten using Equation 9.17 and utilizing the definition of the discretized contact interface as follows:

$$\hat{\mathbf{f}}_a^{(1)} = \sum_{\alpha \in \mathcal{W}_a} \int_{\omega_\alpha} \hat{p}_\alpha \bar{\mathbf{n}}_\alpha \; \bar{N}_a^{(1)} da, \tag{9.21}$$

and

$$\hat{\mathbf{f}}_b^{(2)} = -\sum_{\alpha \in \mathcal{W}_b} \int_{\omega_\alpha} \hat{p}_\alpha \bar{\mathbf{n}}_\alpha \; \bar{N}_b^{(2)} da, \tag{9.22}$$

where $\mathcal{W}_c$ is the set of intermediate surfaces, ω_α, that contribute to the contact nodal force at node c, which is implied to lie on side 1 or 2. Note that this work focuses on *normal* contact enforcement. As a result, we proceed only with the normal component of the contact traction. The contact nodal forces rewritten in Equations 9.21 and 9.22 do not contain a subscript or superscript denoting the normal component. Rather, it is assumed for clarity that the normal component is being handled heretofore unless otherwise noted. We proceed

by plugging Equation 9.20 into Equations 9.21 and 9.22, which gives

$$\hat{\mathbf{f}}_a^{(1)} = \sum_{\alpha \in \mathcal{W}_a} \int_{\omega_\alpha} (q_0^\alpha + r_1^\alpha x + r_2^\alpha y)\ \bar{\mathbf{n}}_\alpha\ \bar{N}_a^{(1)} da, \tag{9.23}$$

and

$$\hat{\mathbf{f}}_b^{(2)} = -\sum_{\alpha \in \mathcal{W}_b} \int_{\omega_\alpha} (q_0^\alpha + r_1^\alpha x + r_2^\alpha y)\ \bar{\mathbf{n}}_\alpha\ \bar{N}_b^{(2)} da, \tag{9.24}$$

More compactly, Equations 9.23 and 9.24 can collectively be expressed

$$\hat{\mathbf{f}}_a^{(i)} = \sum_{\alpha \in \mathcal{W}_a} \left(\bar{\mathbf{n}}_\alpha \int_{\omega_\alpha} \bar{N}_a^{(i)} da\ q_0^\alpha + \bar{\mathbf{n}}_\alpha \int_{\omega_\alpha} \bar{N}_a^{(i)} x da\ r_1^\alpha + \bar{\mathbf{n}}_\alpha \int_{\omega_\alpha} \bar{N}_a^{(i)} y da\ r_2^\alpha \right), \tag{9.25}$$

where the negative sign for $i = 2$ is absorbed into the coefficients. Equation 9.25 results in the following two linear systems:

$$\hat{\mathbf{f}}^{(1)} = \mathbf{M}^{(1)} \mathbf{p}, \tag{9.26}$$

$$\hat{\mathbf{f}}^{(2)} = -\mathbf{M}^{(2)} \mathbf{p}, \tag{9.27}$$

where $\mathbf{p}$ is the global vector of pressure coefficients. Equations 9.26 and 9.27 may be written in a compact form similar to Equation 9.25 as

$$\hat{\mathbf{f}}^{(i)} = \mathbf{M}^{(i)} \mathbf{p},\ i = 1, 2. \tag{9.28}$$

The vectors $\hat{\mathbf{f}}^{(1)} \in \mathbb{R}^{(nsd)*m}$ and $\hat{\mathbf{f}}^{(2)} \in \mathbb{R}^{(nsd)*n}$ are the global vectors of contact nodal forces for sides 1 and 2, respectively. The dimension of these vectors corresponds to the number of spatial dimensions (nsd) multplied by the total number of nodes associated with contacting face-pairs, i.e. m and n, for sides 1 and 2, respectively. Let the integers $nsd * m$ and $nsd * n$ be denoted $\tilde{m}$ and $\tilde{n}$. Then, $\mathbf{M}^{(1)} \in \mathbb{R}^{\tilde{m} \mathrm{x} (nsd)N}$ and $\mathbf{M}^{(2)} \in \mathbb{R}^{\tilde{n} \mathrm{x} (nsd)N}$ are linear operators that map a global vector of contact pressure coefficients, $\mathbf{p} \in \mathbb{R}^{(nsd)N}$, to contact nodal forces, where there are three unknown linear pressure coefficients for each of the total N

intermediate surfaces in 3D and two unknown pressure coefficients in 2D. For simplicity, let us denote the number of columns in $\mathbf{M}^{(i)}$ as $\tilde{N}$.

The elements of the linear operator $\mathbf{M}^{(i)}$ can be written as

$$m_{0a}^{(i)\alpha} = a_\alpha \bar{\mathbf{n}}_\alpha \int_{\omega_\alpha} \bar{N}_a^{(i)} da, \tag{9.29}$$

$$m_{1a}^{(i)\alpha} = a_\alpha \bar{\mathbf{n}}_\alpha \int_{\omega_\alpha} \bar{N}_a^{(i)} x da, \tag{9.30}$$

and

$$m_{2a}^{(i)\alpha} = a_\alpha \bar{\mathbf{n}}_\alpha \int_{\omega_\alpha} \bar{N}_a^{(i)} y da, \tag{9.31}$$

where

$$a_\alpha = \begin{cases} 1, & \text{if } a \in \mathcal{A}_\alpha^{(i)}, \\ 0, & \text{otherwise.} \end{cases} \tag{9.32}$$

The set $\mathcal{A}_\alpha^{(i)}$ is the set of nodes belonging to the face on the i^{th} contact surface that is used to form ω_α. Said another way, $\mathcal{A}_\alpha^{(i)}$ is the set of nodes on the i^{th} contact surface that recieve contact nodal force contributions from ω_α. Equation 9.33 shows Equation 9.28 in expanded form.

$$\begin{Bmatrix} \hat{\mathbf{f}}_1^{(i)} \\ \hat{\mathbf{f}}_2^{(i)} \\ \vdots \\ \vdots \\ \vdots \\ \hat{\mathbf{f}}_r^{(i)} \end{Bmatrix} = \begin{bmatrix} m_{01}^{(i)1} & m_{11}^{(i)1} & m_{21}^{(i)1} & \dots & \dots & m_{01}^{(i)N} & m_{11}^{(i)N} & m_{21}^{(i)N} \\ m_{02}^{(i)1} & m_{12}^{(i)1} & m_{22}^{(i)1} & \dots & \dots & m_{02}^{(i)N} & m_{12}^{(i)N} & m_{22}^{(i)N} \\ \vdots & \vdots & \vdots & \ddots & & \vdots & \vdots & \vdots \\ \vdots & & & & & & & \vdots \\ \vdots & \vdots & \vdots & & \ddots & \vdots & \vdots & \vdots \\ m_{0r}^{(i)1} & m_{1r}^{(i)1} & m_{2r}^{(i)1} & \dots & \dots & m_{0r}^{(i)N} & m_{1r}^{(i)N} & m_{2r}^{(i)N} \end{bmatrix} \begin{Bmatrix} q_0^1 \\ r_1^1 \\ r_2^1 \\ \vdots \\ q_0^N \\ r_1^N \\ r_2^N \end{Bmatrix}. \tag{9.33}$$

It is important to note that the majority of the entries in $\mathbf{M}^{(i)}$ shown in Equation 9.33 are zero. In general, $\mathbf{M}^{(i)}$ is sparse. Equation 9.33 shows the general structure of the entries, though most m contributions will be zero because a particular ω_α will only fall under the support of four (2D) or eight (3D) finite element nodal basis functions.

Equations 9.26 and 9.27 play a significant role in this work. Rather than interpolating the unknown traction field using the nodal shape functions, as is done in a mortar method, we make the observation that the *same* global contact pressure coefficients, the elements of $\mathbf{p}$, are mapped by each $\mathbf{M}^{(i)}$ to global contact nodal force vectors for each contact surface. In fact, for the contact problem between two deformable bodies, the *global* contact nodal force vector may be expressed as a linear mapping of pressure coefficients:

$$\hat{\mathbf{f}} = \mathbf{M}\mathbf{p}, \tag{9.34}$$

where $\hat{\mathbf{f}}$ is

$$\hat{\mathbf{f}} = \left[\begin{array}{c} \hat{\mathbf{f}}^{(1)} \\ \hline \hat{\mathbf{f}}^{(2)} \end{array}\right] \in \mathbb{R}^{(\tilde{m}+\tilde{n})}, \tag{9.35}$$

and $\mathbf{M}$ is

$$\mathbf{M} = \left[\begin{array}{c} \mathbf{M}^{(1)} \\ \hline \mathbf{M}^{(2)} \end{array}\right] \in \mathbb{R}^{\tilde{M}\mathrm{x}\tilde{N}}. \tag{9.36}$$

The row dimension, $\tilde{M}$, is now used to denote the total number of contact nodes. $\mathbf{p} \in \mathbb{R}^{\tilde{N}}$ is the usual vector of pressure coefficients. Note that $\tilde{M} > \tilde{N}$ for practical finite element meshes; as a result, we require $\mathbf{M}$ to be of full column rank. This ensures a unique mapping of $\mathbf{p} \rightarrow \hat{\mathbf{f}}$ where $ker\{A\} = \emptyset$. There is no guarantee, however, that $\mathbf{M}$ will be of full column rank. In fact, depending on the mesh densities of $\Gamma_c^{(1)}$ and $\Gamma_c^{(2)}$, $\mathbf{M}$ may have a degree of indeterminacy, leading to non-unique solutions for $\mathbf{p}$. What this means, practically speaking, is that more overlaps exist than are necessary to determine a unique set of nodal

forces given a setting of pressure coefficients. The overconstraint of $\mathbf{M}$ is more likely to occur when the mesh densities of $\Gamma_c^{(1)}$ and $\Gamma_c^{(2)}$ are sufficiently different. Rather than impose mesh restrictions, the overconstraint can be mitigated by ensuring that $\mathbf{M}$ is of full column rank.

Let us assume that we have a full-column rank operator mapping contact pressure coefficients to contact nodal forces. Let us denote such an operator as $\hat{\mathbf{M}}$. We may write this mapping as

$$\hat{\mathbf{f}} = \hat{\mathbf{M}}\hat{\mathbf{p}}, \tag{9.37}$$

where $\hat{\mathbf{p}}$ represents the modified vector of pressure unknowns corresponding to the modified $\hat{\mathbf{M}}$. Equation 9.37 is the mapping required to ensure a unique solution of contact nodal forces. One easy way to ensure this is to use the constant pressure terms in the linear polynomial only. That is, if we construct $\mathbf{M}$ using a piecewise constant pressure basis locally at each overlap, then we only populate $\mathbf{M}$ with columns pertaining to the constant term. This automatically results in a full column rank $\hat{\mathbf{M}}$ and for most practical problems, is sufficient in resolving a contact interaction. This point is elaborated upon in Section 9.3.2, while a more general method of *constructing* a full column rank $\hat{\mathbf{M}}$ is outlined in Section 9.3.1.

9.3 Constructing a Full Column Rank Pressure Basis

The operator, $\mathbf{M}$, contains columns associated with the constant and linear terms in the polynomial expression for the uknown contact pressures (see Equation 9.19). Simply utilizing the full $\mathbf{M}$, whose columns form a pressure basis, will often result in indeterminacy in the contact nodal forces. In this case, mathematically, $\mathbf{M}$ is not of full column rank and the pressure basis is not an actual basis, as such. In this work, we construct a full column rank $\mathbf{M}$, herein denoted as $\hat{\mathbf{M}}$, by using a modified Gram-Schmidt method. In doing so, we form $\hat{\mathbf{M}}$, whose columns are the most linearly independent columns of $\mathbf{M}$. As a result, the modified $\hat{\mathbf{p}}$ in Equation 9.37 only contains those terms corresponding to the column vectors of $\mathbf{M}$ that are used to form $\hat{\mathbf{M}}$. The resulting global pressure basis is in fact a basis of $\mathbf{M}$'s column space and comprises the column vectors of $\hat{\mathbf{M}}$ that provide a unique mapping of now active pressure coefficients to contact nodal forces.

9.3.1 Modified Gram-Schmidt Basis Construction

We may use a modified Gram-Schmidt method to construct a pressure basis by isolating linearly independent column vectors of $\mathbf{M}$. The usual Gram-Schmidt method constructs an orthonormal basis amongst a set of input vectors by traversing the vectors, following their initial ordering (see [24]). A *modified* Gram-Schmidt traverses the input set of vectors according to some criteria. We *isolate*, or give preference to, candidate vectors that are largest in magnitude amongst all column vectors of $\mathbf{M}$ that have not yet been previously selected for the pressure basis. Once a candidate vector has been selected, it is orthonormalized with respect to the current basis through a series of successive projections onto the current set of basis vectors. If the resulting candidate vector has a magnitude greater than some user-specified tolerance, it is included in the set of pressure basis vectors; if not, it is discarded and the algorithm moves on to the next candidate vector in $\mathbf{M}$. A psuedo-code algorithm is

outlined below for the modified Gram-Schmidt method.

Algorithm 1 : Construction of $\hat{\mathbf{M}}$ using Modified Gram-Schmidt

```
Input: M ∈ ℝ^{M̃ x Ñ}
initialize M̂
for j = 1 to number of columns in M do
    if j = 1 then
        initialize pressure basis:
        choose column from set of input vectors with largest magnitude,
        cycle
    else
        candidate vector selection:
        select the remaining vector from the input set with largest magnitude
    end if
    perform Gram-Schmidt orthonormalization of j^th candidate vector
    (projection onto current selection of basis vectors)
    check remaining magnitude of j^th candidate vector against specified tolerance
    if magnitude > tolerance then
        include candidate vector in pressure basis
    else
        discard candidate vector
        cycle
    end if
end for
```

Strictly speaking, we do not use the orthonormal basis that is the result of the Gram-Schmidt algorithm applied to the set of column vectors of $\mathbf{M}$. In fact, we do not require a basis of the column space of $\mathbf{M}$ be orthonormal. Moreover, the orthonormal basis vectors provided by the Gram-Schmidt algorithm may not preserve the sparsity of the original $\mathbf{M}$ operator, which is desirable in order to preserve the sparsity of the global equations. Having said that, we need a method to determine which column vectors of $\mathbf{M}$ may be used to construct a basis of the column space of $\mathbf{M}$. Said another way, we need to know which columns of $\mathbf{M}$ are linearly independent, the collection of which span the column space of $\mathbf{M}$. The Gram-Schmidt method gives us this information. Though somewhat computationally inefficient, the modified Gram-Schmidt method implemented in this work will return a list of column indices indicating the exact input vectors that *would* be used to construct an orthonormal

basis. That is, the method selects the most linearly independent vectors of the given input set of vectors. This is the exact information we need in selecting column vectors of $\mathbf{M}$ to form a basis of the column space of $\mathbf{M}$, held in the modified pressure mapping, $\hat{\mathbf{M}}$. Though this process modifies the active pressure coefficients based on the selected columns of $\mathbf{M}$, this process guarantees a unique solution for the contact nodal forces and preserves sparsity in the system of equations.

9.3.2 Using a Constant Pressure Basis

The forgoing introduces the linear operator, $\mathbf{M}$, that maps pressure coefficients to contact nodal forces. We have discussed the notion of a pressure basis, which is in fact a basis of the column space of $\mathbf{M}$. If a sufficiently rich basis is used, then $\mathbf{M}$ is no longer of full column rank itself, and the construction of a full column rank $\hat{\mathbf{M}}$ operator whose columns serve as a basis of the column space of $\mathbf{M}$ is required. The notion of a pressure basis and a full column rank mapping of pressure coefficients to contact nodal forces is not discussed in the literature in great detail. One likely reason is that many contact formulations choose to represent the contact traction over the discretized interface as a collection of piecewise constant pressures. That is, if a constant pressure basis is used, a full column rank $\mathbf{M}$ is in fact provided and $\hat{\mathbf{M}} = \mathbf{M}$. It must be noted, however, that for a fundamental problem involving stacked, two block contact with a *linearly* distributed load on the top surface of the top block, such as that shown in (a) in Figure 18, a constant pressure basis is unable to resolve the contact interaction resulting in a state of equilibrium. A state of equilibrium results from a linear contact pressure, as shown in (b) in Figure 18. A uniformly distributed contact load (i.e. a constant contact pressure) shown in (c) in Figure 18 results in an unresolved moment on the top block when used to resist the linearly distributed load. That is, there is no moment equilibrium on the top block when the linearly distributed load is resisted by a constant contact pressure. This results in an unresolved moment, diagramatically shown as $f_r * e$, effectively induced from the linear load's resultant force acting eccentrically from center, as shown in (c). The unresolved moment may equivalently be expressed as a lateral load acting at the top of the top block, shown as f_{top} in Figure 18 (d). The lack of moment equilibrium on the top block will result in unrestrained sliding of the top block. This fundamental "error" in using a constant pressure basis exists mathematically, but may be mitigated by mesh refinement. That is, rather than a single block-on-block contact problem, each block may be discretized into multiple elements.

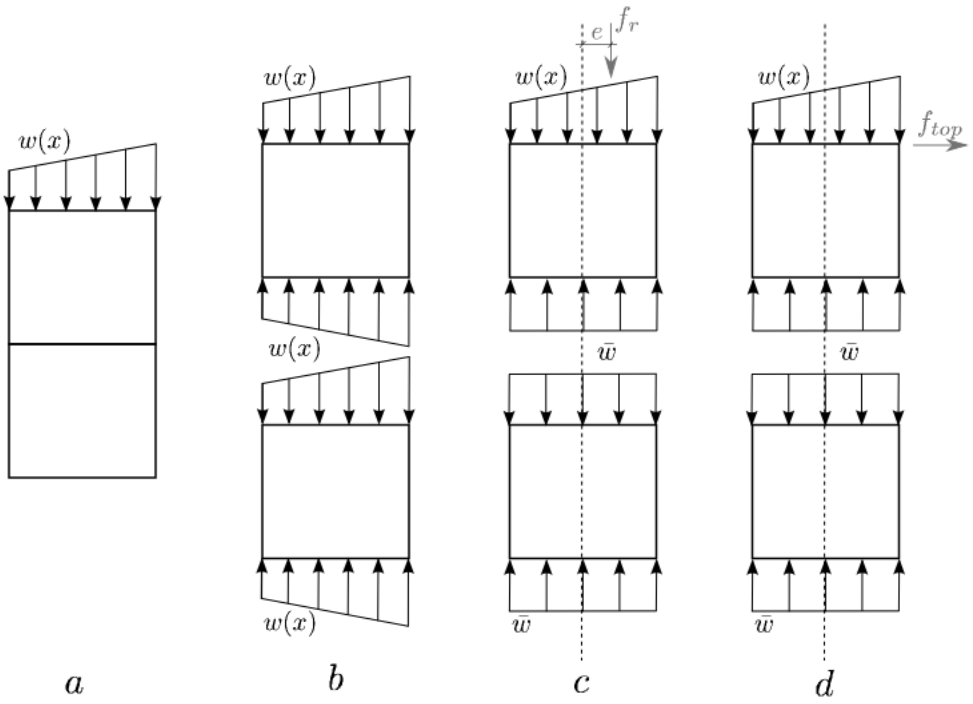

Figure 18: (a) A stacked, two block contact problem with a linearly distributed load on the top block, (b) an equilibrium force diagram where the ideal contact distribution is an equal-and-opposite *linearly* distributed load, (c) free body diagram where contact is enforced using a *uniformly* distributed load, as would be the case with a constant pressure basis. The force, f_r, denotes the resultant force of the linearly distributed load, which is applied at an eccentricity, e, from center. This is noted in gray to imply that an unresolved moment acts on the top block. (d) The force, f_{top}, shown in gray is meant to represent an alternative diagramatic expression of this moment, indicating that an unresolved equivalent lateral force acts on the top block.

Numerical experimentation has shown that a constant pressure basis is adequate to resolve most contact problems. As a result, solely a constant pressure basis is used henceforth in this work. This obviates the need, algorithmically, to construct a $\hat{\mathbf{M}}$ operator, and does in fact yield a simpler formulation. The question still remains, however, what is to be gained from a richer pressure basis? While the answer to this may be linked to the exact types of surface geometry that are being used to create facet-pairs and contact overlaps, two obvious conclusions come to mind. Having the control to use a linear pressure basis allows greater flexibility in enforcing contact over either difficult interfaces or under-resolved interfaces, as is the case in the problem presented in Figure 18. Using the problem presented in Figure 18

as an example, a linear pressure basis may be able to resolve a contact interaction whereas using a constant pressure basis would have required mesh refinement. Ultimately, a richer pressure basis may result in an algorithm that is less sensitive to interface geometry and mesh resolution. Another reason one may wish to use a linear pressure basis, though this is an open area of research, is for the ability to control the basis itself and which contact overlaps play a significant role. In fact, the modified Gram-Schmidt algorithm discussed in the previous section allows control over which overlaps are given preference in constructing a basis. Numerical experimentation has shown that prioritizing candidate basis vectors based on their magnitude will give preference to larger areas of overlap. The implications of this are not fully understood, but this control may yield a more flexible and robust contact algorithm. On the other hand, some care must be taken when constructing "too rich" a basis as one may encounter an inf-sup-type instability in the pressure distribution. Early numerical work using a linear pressure basis demonstrated a sensitivity to this type of instability in the pressure solution. This motivated the development and use of the modified Gram-Schmidt method for constructing a full column rank linear operator, which precludes the instability in the solution, while still allowing one to use a "richer" pressure basis.

Though a constant pressure basis is used in this work, the mathematical presentation is done considering a full linear pressure basis, with simplifications associated with a constant basis explicitly noted where appropriate.

9.3.3 Notes on the Linearization of Contact Residual Contributions

Let the total derivative of Equation 9.6, written

$$D\mathbf{R}(\hat{\mathbf{u}},\hat{\mathbf{p}}) = \frac{\partial \mathbf{R}}{\partial \hat{\mathbf{u}}} + \frac{\partial \mathbf{R}}{\partial \hat{\mathbf{p}}} = \frac{\partial \bar{\mathbf{R}}}{\partial \hat{\mathbf{u}}} + \frac{\partial \bar{\mathbf{R}}}{\partial \hat{\mathbf{p}}} + \frac{\partial \hat{\mathbf{R}}}{\partial \hat{\mathbf{u}}} + \frac{\partial \hat{\mathbf{R}}}{\partial \hat{\mathbf{p}}}, \tag{9.38}$$

represent the tangent stiffness contributions of the finite element residual, where $\hat{\mathbf{u}}$ and $\hat{\mathbf{p}}$ are the global incremental displacement unknowns and the contact pressure coefficient unknowns, respectively. An obvious note is that $\frac{\partial \bar{\mathbf{R}}}{\partial \hat{\mathbf{p}}} = \mathbf{0}$ because $\bar{\mathbf{R}}$ by its very definition does not depend on $\hat{\mathbf{p}}$. Furthermore, $\frac{\partial \bar{\mathbf{R}}}{\partial \hat{\mathbf{u}}}$ is the standard tangent stiffness arising from the solid mechanics variational boundary value problem *without* contact. Therefore, implementing contact into an existing finite element code means that this portion of Equation 9.38 is in place. As a result, let us address $\frac{\partial \hat{\mathbf{R}}}{\partial \hat{\mathbf{u}}}$ and $\frac{\partial \hat{\mathbf{R}}}{\partial \hat{\mathbf{p}}}$.

In the general setting, the contact contribution to the residual in Equation 9.8 is expressed again below in the current configuration,

$$\hat{R}_{ja} = -\int_{\gamma_c^{(i)}} \hat{t}_j^{(i)} \varphi_a^{(i)} da, \;\; i = 1, 2. \tag{9.39}$$

Using Equation 9.20 in Equation 9.39, along with the discretization of the contact interface, yields

$$\hat{R}_{ja} = -\sum_{\alpha \in \mathcal{W}_a} \bar{n}_{j\alpha} \left(\int_{\omega_\alpha} \bar{N}_a^{(i)} da \; q_0^\alpha + \int_{\omega_\alpha} x \bar{N}_a^{(i)} da \; r_1^\alpha + \int_{\omega_\alpha} y \bar{N}_a^{(i)} da \; r_2^\alpha \right), \tag{9.40}$$

which can be rewritten as

$$\hat{R}_{ja} = -\sum_{\alpha \in \mathcal{W}_a} \bar{n}_{j\alpha} \left(\int_{\omega_\alpha} \psi_1 \bar{N}_a^{(i)} da \; q_0^\alpha + \int_{\omega_\alpha} \psi_2 \bar{N}_a^{(i)} da \; r_1^\alpha + \int_{\omega_\alpha} \psi_3 \bar{N}_a^{(i)} da \; r_2^\alpha \right). \tag{9.41}$$

Here $\psi = \{1, x, y\}$ can be viewed as a basis for $\hat{p}_\alpha$, where $\psi_c, c = 1, 2, 3$, denotes an element of the set. Equation 9.41 corresponds to Equation 9.34, which again arises in the general setting. This work, however, pursues a modified linear system of equations for the contact nodal forces represented by Equation 9.37, which eliminates indeterminacy in the contact pressure coefficients. As a result, while in the general setting $\psi_m, m = 1, 2, 3$ represents the three columns of $\mathbf{M}$ per ω_α, $\hat{\mathbf{M}}$ will be constructed with a selection of columns where the three columns for a given ω_α may be omitted entirely or a subset of ψ will appear in the equations. As a consequence, Equation 9.41 is modified as

$$\hat{R}_{ja} = -\sum_{\alpha \in \mathcal{W}_a} \sum_{c \in \mathcal{C}_\alpha} \bar{n}_{j\alpha} \int_{\omega_\alpha} \psi_c^\alpha \bar{N}_a^{(i)} da \, p_c^\alpha, \tag{9.42}$$

where $\mathcal{C}_\alpha$ is the set of $\psi_c, c = 1, 2,$ or 3 that pertain to the columns of $\mathbf{M}$ that are retained in $\hat{\mathbf{M}}$ for a given overlap, ω_α; and p_c^α are the corresponding pressure coefficients associated with $\psi_c, c \in \mathcal{C}_\alpha$. Equation 9.42 is then equivalent to Equation 9.37.

A very important note when considering the total derivative contribution, $\frac{\partial \hat{\mathbf{R}}}{\partial \hat{\mathbf{u}}}$, is that this work considers the domain of integration, ω_α, and therefore $\bar{\mathbf{n}}_\alpha$, *fixed* for a given Newton iteration. As a result, $\frac{\partial \hat{\mathbf{R}}}{\partial \hat{\mathbf{u}}} = \mathbf{0}$. In general, the displacements of the nodes belonging to a given face-face interaction will change in a given Newton iteration. As a result, a $k+1$ solution for $\hat{\boldsymbol{\tau}}^{(n)}$ will be associated with ω_α regions over which $\hat{\boldsymbol{\tau}}^{(n)}$ acts that were formed using the k^{th} iterate's current configuration. What this means is that we will lose some symmetry between $k+1$ current configuration of the face-face interaction and the iteration-k ω_α, which results in a small deviation from angular momentum conservation. As the two faces in a local interaction become co-planar, then angular momentum will be conserved.

This leaves us to determine $\frac{\partial \hat{\mathbf{R}}}{\partial \hat{\mathbf{p}}}$. Using Equation 9.42, we can write

$$\frac{\partial \hat{R}_{ja}}{\partial \hat{p}_k} = -\bar{n}_{j\alpha} \int_{\omega_\alpha} \psi_k^\alpha \bar{N}_a^{(i)} da \; \forall \; \alpha \in \mathcal{W}_a, k \in \mathcal{C}_\alpha, \tag{9.43}$$

which is in fact an element of the $\hat{\mathbf{M}}$ operator. Therefore, we may write,

$$\frac{\partial \hat{\mathbf{R}}}{\partial \hat{\mathbf{p}}} = \hat{\mathbf{M}}, \tag{9.44}$$

where in component form, $\frac{\partial \hat{R}_{ja}}{\partial \hat{p}_k} = [m]_{jak\alpha}$. The range of the first two indices is $j = 1, 2, 3$ and $a = 1$ to number of finite element nodes included in contact facet-pairs. These indices refer to a row in $\hat{\mathbf{M}}$ corresponding to a spatial component j for a given node a. The second pair of indices refer to a column in $\hat{\mathbf{M}}$. Index k for a given index α refers to the active pressure coefficient associated with the ω_α overlap. Equation 9.44 is the tangent stiffness contribution from the contact residual term.

9.4 Kinematic Constraint and the Active Set

The kinematic constraint is used to enforce contact, and may be expressed, in very general terms, as

$$g \leq 0, \tag{9.45}$$

the weak form of this inequality will be presented later in this section. Equation 9.45 is often referred to as the contact inequality constraints, featuring the scalar value *gap* function, g. If a closest point projection method is used to form the inequality constraint then g takes the form shown in Equation 6.2. This work uses an integral form of the gap function, which will be presented later in this section.

9.4.1 Introduction to the Active Set

In order to enforce Equation 9.45, one must identify all face-face pairs with kinematically inadmissable configurations[2]. Doing so allows us to construct the contact interface, γ_c, from all intermediate surfaces, ω_α, $\alpha = 1, N$, which are in turn constructed from $\{f_j^{(1)}, f_k^{(2)}\}$ face pairs of this kinematically inadmissable set. To proceed, the active set of face-face pairs is denoted,

$$\mathcal{A}_c^f = \left\{ \{f_j^{(1)}, f_k^{(2)}\}, \{f_m^{(1)}, f_n^{(2)}\}, ..., \ \{f_p^{(1)}, f_q^{(2)}\} \right\}, \tag{9.46}$$

where the subscript on $\mathcal{A}$ refers to a unique face index and the superscript on $\mathcal{A}$ refers to the contact surface to which the face belongs. Since every face-face pair results in an intermediate surface, ω_α, we may also express the active set in terms of the set of intermediate surfaces as

$$\mathcal{A}_c^\omega = \{\omega_1, \omega_2, ..., \omega_N\}. \tag{9.47}$$

The following section discusses the procedure for evaluating the active set of contacting faces.

9.4.2 Collision Detection and the Active Set

The active set is determined through a series of refined proximity detection methods. First, a rough proximity screening takes place to determine which faces on side 1 are *close* to which faces on side 2. In this work, the radius of the smallest sphere (circle in 2D) that contains a face is determined for each face on $\gamma_c^{(1)}$ and $\gamma_c^{(2)}$. If the distance between the centroids of two faces (one on side 1 and the other on side 2) is less than the sum of the aforementioned radii, then the two faces form a face-face interaction *candidate*. This is an $\mathcal{O}(n^2)$ operation, though with very low level calculations using minimal stored data. This rough proximity

[2]The point at which kinematically inadmissable configurations are checked is a matter of algorithm design and will be discussed in Section 10

search can be made more efficient with a bounding box type method in which face-face checks are performed only between faces in respective bounding boxes. The optimization of the contact search was not a priority in this work and is an established subject matter in the literature.

Once potential face-face interaction candidates have been identified, one of two algorithmic choices are proposed. In this work, the gap between the two candidate faces is computed and if interpenetration occurs, or nearly occurs up to some tolerance, then a median plane is constructed per Equation 8.1. Then, the two faces are projected onto the median plane and their intersection computed. Faces with nonzero intersecting face projections are thrown into the active set. An alternative method that is proposed in lieu of the aforementioned process is a collision detection algorithm.

A collision detection algorithm may be used to determine if the two faces intersect, thus forming a local contact interaction. The collision detection algorithm proposed here not only determines the minimum distance of separation or the maximal distance of interpenetration of two polytopes, but also provides a unit normal defining a median plane characteristic to the geometric separation or interpenetration. The input required is simply the vertex point sets of the current configuration of two faces under consideration. If the detection algorithm returns a maximal distance of interpenetration, we include the face-face interaction in the active set.

The collision detection method is formulated in the following way. Consider Figure 19, which shows two polygons, A and B, with vertices $\mathbf{a_i}$ and $\mathbf{b}_j$, respectively. We define the minimal distance between the polygons as

$$\bar{d} = \min_{\mathbf{x},\mathbf{y}} |\mathbf{x} - \mathbf{y}|, \ \forall \, \mathbf{x} \in A, \mathbf{y} \in B. \tag{9.48}$$

Equation 9.48 is the problem we would like to solve. Doing so gives the minimum distance

of separation or maximum interpenetration between two convex hulls. To go about solving Equation 9.48, let us again refer to Figure 19 and define a plane s with normal $\mathbf{n}$ that, in a loose sense, is shown to "separate" polygons A and B. Using this surface of separation, we may define the following,

$$d_{ij} = \mathbf{n} \cdot \mathbf{a}_i - \mathbf{n} \cdot \mathbf{b}_j, \tag{9.49}$$

as the difference in the scalar projections of $\mathbf{a}_i$ and $\mathbf{b}_j$ onto $\mathbf{n}$. Let us further define

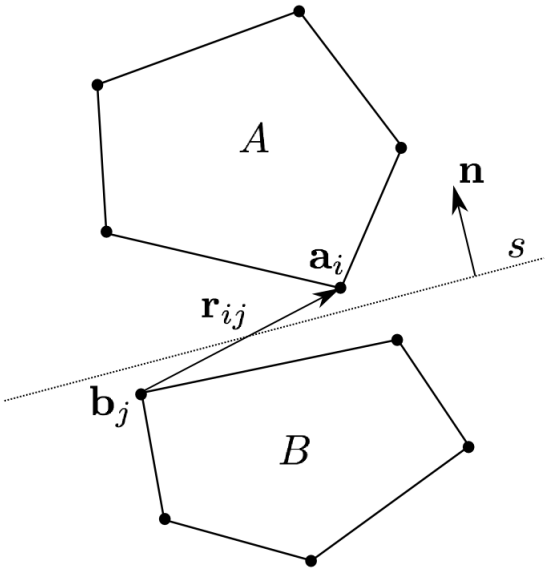

Figure 19: The collision detection parameters are shown for two n-gons. The vertices of polygon A and polygon B, denoted $\{\mathbf{a}_i\}$ and $\{\mathbf{b}_j\}$, respectively, are the input point set data, while $\mathbf{n}$ is the normal to plane s, which helps to form the minimal distance relation.

$$d = \min_{i,j}(d_{ij}), \tag{9.50}$$

which is the minimum difference in projections. Using Equation 9.49 in Equation 9.50 we obtain

$$d = \min_{i}(\mathbf{n} \cdot \mathbf{a}_i) + \min_{j}(\mathbf{n} \cdot -\mathbf{b}_j), \tag{9.51}$$

which may be rewritten as

$$d = \min_{i,j}\big(\mathbf{n} \cdot (\mathbf{a}_i - \mathbf{b}_j)\big) = \min_{i,j}(\mathbf{n} \cdot \mathbf{r}_{ij}), \tag{9.52}$$

where $\mathbf{r}_{ij} = (\mathbf{a}_i - \mathbf{b}_j)$ is referred to as a "link-vector", linking vectors $\mathbf{a}_i$ and $\mathbf{b}_j$. The crux of the algorithm is in fact to determine the optimal $\mathbf{n}$ that gives us the minimal distance of separation while minimizing the dot product with the link vector, $\mathbf{r}_{ij}$ shown in Equation 9.52.

To motivate the use of the separation plane, s, and its unit normal, $\mathbf{n}$, in determining the minimal separation distance, $\bar{d}$, let us consider Figure 20 where the minimal distance between A and B is between the actual vertices $\mathbf{a}_i$ and $\mathbf{b}_j$ as shown. In order to find the minimal

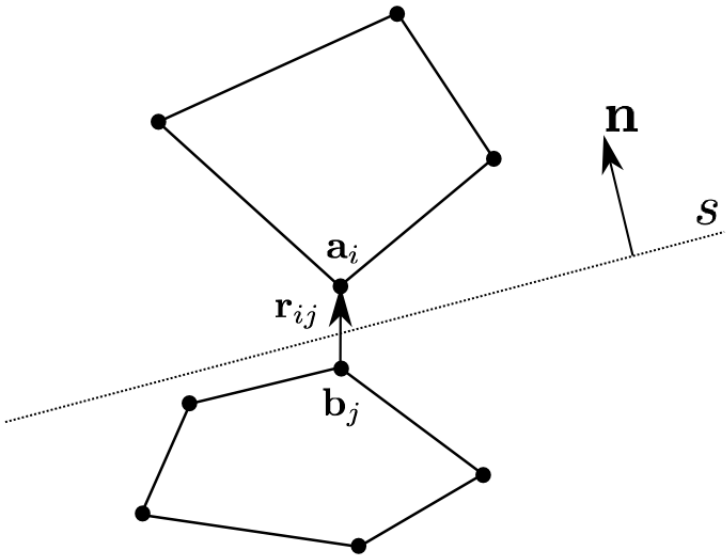

Figure 20: The collision detection setup for two polygons whose minimal distance occurs between $\mathbf{a}_i$ and $\mathbf{b}_i$

distance of separation, or the maximal distance of interpenetration, however, we must place restrictions on $\mathbf{n}$. To do so, consider Figure 21, which shows the link vector, $\mathbf{r}_{ij}$ projected onto $\mathbf{n}$ belonging to an arbitrary separation plane, s, as shown. Let the vectors $\mathbf{p}$ and $\mathbf{q}$ in

Figure 21 be defined as

$$\mathbf{p} = (\mathbf{n} \cdot \mathbf{r}_{ij})\mathbf{n}, \tag{9.53}$$

and

$$\mathbf{q} = \mathbf{r}_{ij} - (\mathbf{n} \cdot \mathbf{r}_{ij})\mathbf{n}. \tag{9.54}$$

Let us assume, for the moment, that $\mathbf{r}_{ij} \cdot \mathbf{n} = |\mathbf{r}_{ij}|$, where $|\mathbf{n}| = 1$. That is, $\mathbf{n}$ is unit and colinear with $\mathbf{r}_{ij}$. Then Equation 9.52 yields the minimal distance of separation between A and B, which is the expected result. In the general case, however, the orientation of s is not

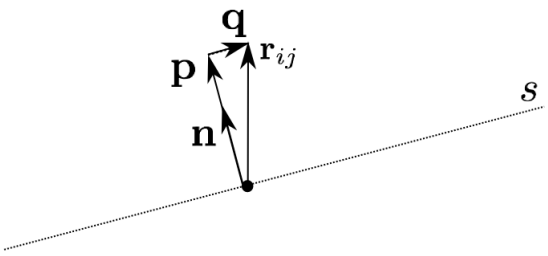

Figure 21: The relation between the normal, $\mathbf{n}$, to plane s and a link vector, $\mathbf{r}_{ij}$.

know a priori. Considering Figure 21, we see that $\mathbf{r}_{ij} \cdot \mathbf{n}$ is maximized when $\mathbf{n}$ is such that $\mathbf{r}_{ij} \cdot \mathbf{n} = |\mathbf{r}_{ij}|$. This means that the vectors $\mathbf{p}$ will be as large in magnitude as possible and $\mathbf{q}$'s will be as small as possible when this condition is met. For the example shown in Figure 20, the optimal $\mathbf{n}$, which yields the maximum $\mathbf{r}_{ij} \cdot \mathbf{n}$, results in $\mathbf{p} = \mathbf{r}_{ij}$ and $\mathbf{q} = \mathbf{0}$. This implies a critical result, that the minimum distance of separation is the one that maximizes $\mathbf{r}_{ij} \cdot \mathbf{n}$ for all possible $\mathbf{n}$. Accordingly, we may define the minimal distance of separation as

$$\bar{d} = \max_{|\mathbf{n}|=1} \min_{i,j} (\mathbf{r}_{ij} \cdot \mathbf{n}). \tag{9.55}$$

For a face-face interaction candidate whose solution to Equation 9.55 yields a maximal distance of interpenetration, which also corresponds to the thickness of the smallest "slab" containing the intersection of $f_j^{(1)}$ and $f_k^{(2)}$, the interaction is then included in the active set.

9.4.3 Perspectives on the Kinematic Constraint and the Active Set

An active set strategy where one identifies all kinematically inadmissable face-pairs is typically used to convert the inequality constraint in Equation 9.45 to an equality constraint, which is easier to enforce. The idea is that if we can identify all kinematically inadmissable face-pairs, then we simply have to enforce zero interpenetration, or $g = 0$, at each of these pairs; thus, alleviating the burden of dealing with inequality constraints.

The notion of converting an inequality constraint to an equality constraint through the use of an active set presupposes that all face-pairs in the kinematically inadmissable set *stay* in contact. That is, no face-pair separates during the contact enforcement. In very general terms, in the implicit solution setting, this means that no face-pair comes *out of* contact given an increment of displacement. The experience gained through this work has shown that this can only be ensured under simplified geometries, simplified deformation paths, or under customized boundary conditions and cannot, in general, be guaranteed.

The consequence of the aforementioned observation is that for a given face-pair in the active set that may come out of contact, a non-physical, tensile contact pressure solution results. Furthermore, if a face-pair *nearly* violates the constraint, but is not included in the active set, then given an increment of displacement that face-pair may exhibit interpenetration as a consequence of its exclusion from the active set. Specifics regarding these observations in the implicit solution setting are outlined in Sections 10 and 11 of this work. A significant conclusion, however, that strongly motivated the direction of this work, is that simply admitting constraint violating pairs into the active set and enforcing a subsequent equality constraint, compromises robustness of a contact methodology under more difficult contact problems.

9.4.4 Inequality to Equality Constraint and the Active Set

The previous section outlined limitations in using an active set to enforce an equality constraint. Using current configuration face-pair geometry is not a good predictor of a deformed geometry given an increment of deformation. As previously explained, interpenetrating face-pairs may want to separate and separated face-pairs may want to interpenetrate. The active set formed based on current configuration geometry presupposes that a kinematically admissable or inadmissable configuration stays this way through an increment of deformation. Early numerical work demonstrated that this was not always the case, the consequences of which are discussed in the previous section of this work. As a result, there was strong motivation to handle the inequality constraint presented in Equation 9.45 directly. While an active set of face-pairs is still formed, it became clear that a method to handle interpenetrating face-pairs that separate, and separated face-pairs that come to interpenetrate given an increment of deformation was needed. As a result, the active set is not only comprised of kinematically *inadmissable* face-pairs, but is extended (and possibly overpopulated) with face-pairs exhibiting separation that *may* come to interpenetrate. A method that handles the inequality constraint directly will appropriately handle interpenetration and separation without nonphysical solutions or missed face-pair interactions. While the numerical details of the method devised in this work to handle the inequality constraints will be discussed in Section 10, the point here is to provide the reader a feel and intuition regarding the practical nuances of the kinematic constraint.

The obvious point at this juncture is how to treat the inequality constraint. Ultimately we want to transform Equation 9.45 to an equality constraint, but must look beyond the active set tool as a way to do this. In this work, we transform Equation 9.45 into an equality constraint by introducing a slack variable, w as follows,

$$c = g - w = 0, \tag{9.56}$$

where c denotes constraint. Equation 9.56 is the final kinematic constraint treated in this contact method. The exact expressions for g and w in Equation 9.56 are presented in the next sections.

9.4.5 The Contact Equality Constraint: Gap Function

To shed light on the exact form of the equality constraint presented in Equation 9.56, let us first consider the gap expression. The gap expression, g, in Equation 9.56, is written as,

$$g(\bar{\mathbf{x}}) = \frac{1}{|\omega_\alpha|} \int_{\gamma_c} (\mathbf{x}^{(1)} - \mathbf{x}^{(2)}) \cdot \mathbf{n} da \quad \forall\, \mathbf{x} \in \gamma_c. \tag{9.57}$$

The gap expression in the constraint Equation 9.56 is a weighted integral expression that weakly enforces zero interpenetration. That is, the use of an integral form enforces zero mean interpenetration over an ω_α. The work presented in [9] noted the smoothing effect this form of the gap expression has in contact problems with large sliding that typically exhibit contact chatter, or are simply unsolvable using node-to-segment methods.

The integral in Equation 9.57 may be written as the sum of integrals over each overlap region, ω_α, which serves as the discretized contact interface:

$$g(\bar{\mathbf{x}}) = \sum_\alpha \frac{1}{|\omega_\alpha|} \int_{\omega_\alpha} (\mathbf{x}^{(1)} - \mathbf{x}^{(2)}) \cdot \bar{\mathbf{n}}_\alpha da \quad \forall\, \bar{\mathbf{x}} \in \omega_\alpha. \tag{9.58}$$

The "weak" form of the gap equation may be formed by multiplying the integrand of Equation 9.58 through by a contact test function as

$$g(\bar{\mathbf{x}}) = \sum_\alpha \frac{1}{|\omega_\alpha|} \int_{\omega_\alpha} (\mathbf{x}^{(1)} - \mathbf{x}^{(2)}) \cdot \bar{\mathbf{n}}_\alpha \psi_\alpha da \quad \forall\, \bar{\mathbf{x}} \in \omega_\alpha, \tag{9.59}$$

where ψ_α is the contact test function.[3] For the contact method presented in this work, the Galerkin expression of the contact test function may be written as

$$\psi_\alpha = \sum_k \zeta_\alpha^k, \tag{9.60}$$

where $\zeta_\alpha^k \in \{1, x, y\}$. That is, there are up to three pressure basis functions deriving from the linear pressure polynomial, Equation 9.19, for each overlap, ω_α. Which of the three pressure basis functions are active per a given overlap is determined from the modification of $\mathbf{M}$ to $\hat{\mathbf{M}}$, outlined in Section 9.3. As previously explained, this work uses a constant pressure basis for each overlap, which means that Equation 9.59 is tested with the constant polynomial term (i.e. $\zeta_\alpha^1 = \zeta = 1$) and reduces to Equation 9.58.

The position vectors in Equation 9.59 permit an isoparametric mapping in the form

$$\mathbf{x}^{(i)}(\boldsymbol{\xi}) = \sum_{a=1}^{nds} N_a^{(i)}(\boldsymbol{\xi}) \mathbf{x}_a^{(i)}, \tag{9.61}$$

where $\boldsymbol{\xi}$ is the vector defining the parent space of the finite element face $f_j^{(i)}$, $N_a^{(i)}$ are the finite element shape functions associated with the face, $\mathbf{x}_a^{(i)}$ are the current configuration position vectors of the finite element nodes on the face, and nds is the number of nodes associated with the finite element face. An important note at this point is that $\bar{\mathbf{x}}$ is defined on ω_α, whereas $\mathbf{x}^{(i)}$ is defined on $\gamma_c^{(i)}$. Specifically, $\mathbf{x}^{(i)}$ is defined on some face, $f_j^{(i)}$, that is used to form ω_α. Considering Equation 9.61, however, shows us that the current configuration position on face $f_j^{(i)}$ is actually a function of $\boldsymbol{\xi}$ defining the face's parent space. To handle this, let us recall the projection operator, $\bar{\mathcal{P}}_\alpha^{(i)} : \bar{\mathbf{x}} \rightarrow \mathbf{x}^{(i)} \; \forall \; \mathbf{x}^{(i)} \in \gamma_c^{(i)}, i = 1, 2$. That is, for every $\bar{\mathbf{x}} \in \omega_\alpha$ we have defined a unique projection onto each $f_j^{(i)}$ used to form ω_α. Additionally,

[3]Note that the undiscretized integral gap function may be written as, $g(\bar{\mathbf{x}}) = \int_{\gamma_c} (\mathbf{x}^{(1)} - \mathbf{x}^{(2)}) \cdot \mathbf{n} \eta_c da$, where η_c is the contact test function and may be written as $\eta_c = \sum_\alpha \sum_k \zeta_\alpha^k$. We may also write the test function as, $\eta_c = \sum_\alpha \psi_\alpha$, where $\psi_\alpha = \sum_k \zeta_\alpha^k$. The discretized integral in Equation 9.59 contains a sum over α, which means the test function in the integrand is simply ψ_α. We have stated that a constant pressure basis is used in this work, so for every ω_α, $\zeta_\alpha^k = 1, k = 1$.

for every $\mathbf{x}^{(i)} \in f_j^{(i)}$ we have a unique $\boldsymbol{\xi}$ determined via an inverse isoparametric mapping of $\mathbf{x}^{(i)}$, the details of which will be discussed in Section 9.5. As a result, we may express the finite element shape function in Equation 9.61 as $N_a(\boldsymbol{\xi}) = N_a(\boldsymbol{\xi}(\bar{\mathcal{P}}_\alpha^{(i)}(\bar{\mathbf{x}})))$. Using this development as well as the expression for the contact test function in Equation 9.60 and a constant pressure basis, we may express Equation 9.59 as

$$g(\bar{\mathbf{x}}) = \sum_\alpha \frac{1}{|\omega_\alpha|} \bar{\mathbf{n}}_\alpha \cdot \int_{\omega_\alpha} \left(\sum_a N_a^{(1)}(\xi(\bar{\mathcal{P}}_\alpha^{(1)}(\bar{\mathbf{x}}))) \, \mathbf{x}_a^{(1)} - \sum_a N_b^{(2)}(\xi(\bar{\mathcal{P}}_\alpha^{(2)}(\bar{\mathbf{x}}))) \, \mathbf{x}_b^{(2)} \right) \zeta \mathrm{da}, \quad \forall \, \bar{\mathbf{x}} \in \omega_\alpha, \zeta = 1. \tag{9.62}$$

Equation 9.62 may be rewritten in the form

$$g(\bar{\mathbf{x}}) = \sum_\alpha \frac{1}{|\omega_\alpha|} \bar{\mathbf{n}}_\alpha \cdot \left(\sum_a \mathbf{x}_a^{(1)} \int_{\omega_\alpha} N_a^{(1)}(\xi(\bar{\mathcal{P}}_\alpha^{(1)}(\bar{\mathbf{x}}))) - \sum_b \mathbf{x}_b^{(2)} \int_{\omega_\alpha} N_b^{(2)}(\xi(\bar{\mathcal{P}}_\alpha^{(2)}(\bar{\mathbf{x}}))) \right) \zeta \mathrm{da}, \quad \forall \, \bar{\mathbf{x}} \in \omega_\alpha, \zeta = 1, \tag{9.63}$$

which clearly shows that the computation of Equation 9.62 ultimately involves evaluations of integrals of finite element shape function over intermediate surfaces, $\omega_\alpha, \alpha = 1, 2, ..., N$. A general form of Equation 9.62 may be written as

$$g(\bar{\mathbf{x}}) = \sum_\alpha \bar{\mathbf{n}}_\alpha \cdot \mathbf{g}_\alpha, \tag{9.64}$$

where

$$\mathbf{g}_\alpha(\bar{\mathbf{x}}) = \int_{\omega_\alpha} \frac{1}{|\omega_\alpha|} \left(\sum_a N_a^{(1)}(\xi(\bar{\mathcal{P}}_\alpha^{(1)}(\bar{\mathbf{x}}))) \, \mathbf{x}_a^{(1)} - \sum_a N_b^{(2)}(\xi(\bar{\mathcal{P}}_\alpha^{(2)}(\bar{\mathbf{x}}))) \, \mathbf{x}_b^{(2)} \right) \zeta \mathrm{da}, \quad \forall \, \bar{\mathbf{x}} \in \omega_\alpha, \zeta = 1. \tag{9.65}$$

Using Equation 9.64 with 9.65 we may write the discrete scalar gap equation as

$$g_\alpha(\bar{\mathbf{x}}) = \bar{\mathbf{n}}_\alpha \cdot \mathbf{g}_\alpha, \tag{9.66}$$

which will be a useful expression when it comes to gap computations and numerical integration over discrete ω_α regions.

9.4.6 The Contact Equality Constraint: Slack Variable

The slack variable, w, in Equation 9.56 is introduced to convert Equation 9.45 to an equality constraint that we may then more easily enforce along side the finite element equilibrium residual. Strictly speaking, the slack variable is an additional, scalar unknown, gap-like quantity. Similar to the gap expression, we test the slack variable term with the constant pressure basis function and integrate over the discretized contact interface,

$$w(\bar{\mathbf{x}}) = \sum_\alpha \int_{\omega_\alpha} w_\alpha \zeta \mathrm{da}, \;\; \forall\, \bar{\mathbf{x}} \in \;\omega_\alpha, \zeta = 1, \tag{9.67}$$

where w_α is the discrete unknown slack variable associated with ω_α. Equation 9.67 may be simplified to

$$w(\bar{\mathbf{x}}) = \sum_\alpha \int_{\omega_\alpha} \mathrm{da}\, w_\alpha, \;\; \forall\, \bar{\mathbf{x}} \in \;\omega_\alpha. \tag{9.68}$$

To scale correctly with Equation 9.66, we multiply Equation 9.68 through by $\frac{1}{|\omega_\alpha|}$, which cancels the area integral in Equation 9.68 resulting in

$$w(\bar{\mathbf{x}}) = \sum_\alpha w_\alpha \;\; \forall\, \bar{\mathbf{x}} \in \;\omega_\alpha. \tag{9.69}$$

The slack variable has been described as an additional scalar valued unknown; however, we must define a pressure-gap relationship in terms of the slack variable in order to pin down

the correct interpenetration/separation constraint enforcement. To do so we introduce a linear pressure-gap relationship.

9.4.7 The $\hat{p}_\alpha - w_\alpha$ Relationship

For each discrete w_α, let us introduce the following linear relation between the discrete unknown contact pressure, p_α, and the associated discrete slack variable, w_α,

$$\alpha_\alpha \hat{p}_\alpha - \beta_\alpha w_\alpha = 0, \tag{9.70}$$

where the $(\alpha_\alpha, \beta_\alpha)$ pair are numerical parameters associated with each ω_α overlap. Notionally, the $(\alpha_\alpha, \beta_\alpha)$ are stiffness parameters controlling a linear spring at each contact overlap. The following equation makes this more apparent:

$$\hat{p}_\alpha = \frac{\beta_\alpha}{\alpha_\alpha} w_\alpha, \tag{9.71}$$

where $\frac{\beta_\alpha}{\alpha_\alpha}$ is the spring "stiffness". As α_α goes to zero, the spring stiffness goes to infinity. As a result, we may take each β_α to scale with a stiffness measure and each $\alpha_\alpha \in [0, 1]$. More precisely, we take each β_α to be

$$\beta_\alpha = (1 - \alpha_\alpha) S_\alpha, \tag{9.72}$$

where for linear elasticity, the characteristic stiffness, S_α, may be taken as the stiffer of the two Young's Moduli of the two opposing contact faces at the ω_α overlap. For J2 plasticity, S_α may be taken as the larger of the two yield stresses. For a Mooney-Rivlin hyperelastic model, S_α may be taken as the larger of the two D_1 parameters, which are related to volumetric response. Numerical experimentation does not show much sensitivity in the exact choice of the S_α parameter, which is an area worth studying. In any case, the forgoing are practical parameter settings based on the material model in use. Note that a spring stiffness parameter

is not actually used numerically. As a result, we do not run the risk of division by a zero α_α parameter. Instead, the $p_\alpha - w_\alpha$ relationship of Equation 9.70 allows us to derive the final form of the discrete kinematic equality constraint.

9.4.8 The Discrete Contact Equality Constraint

The discrete contact equality constraint is written as $g_\alpha - w_\alpha = 0$, where g_α is defined in Equation 9.66 and w_α, though considered a discrete unknown, follows the linear relationship defined in Equation 9.70. To derive the final form of the discrete kinematic constraint we multiply the discrete equality constraint through by β_α and make use of Equation 9.70 to eliminate the w_α unknowns by expressing them in terms of the unknown contact pressures. Thus,

$$c_\alpha = \beta_\alpha g_\alpha - \beta_\alpha w_\alpha = \beta_\alpha g_\alpha - \ \alpha_\alpha \hat{p}_\alpha = 0, \tag{9.73}$$

where c_α generally denotes the contact constraint associated with the overlap ω_α. The driving force behind each discrete constraint expression is the α_α parameter, which ranges from 0 to 1. To understand the effect α_α has on the constraint enforcement, let us consider the upper and lower bound. If $\alpha_\alpha = 1$, then we have $\beta_\alpha = 0$ and

$$\hat{p}_\alpha = 0, \tag{9.74}$$

thereby enforcing a zero pressure constraint. Alternatively, if $\alpha_\alpha = 0$, then we have $\beta_\alpha = S$ and $Sg_\alpha = 0$, which for all $S > 0$ produces

$$g_\alpha = 0, \tag{9.75}$$

thereby enforcing a zero gap constraint. An important note is that by expressing the kinematic constraints per Equation 9.73, we do not at any point risk performing a division by

an α_α or β_α parameter that is set to zero. This is because we do not explicity form or use a contact stiffness parameter. While it is the case that as $\alpha_\alpha \to 0$, the notional contact stiffness, $\frac{\beta_\alpha}{\alpha_\alpha} \to \infty$, and an exact zero gap constraint is enforced, the proposed method in this work does not explicitly utilize a "penalty" like parameter and no divisions by zero occur in practice.

Introducing $(\alpha_\alpha, \beta_\alpha)$ pairs through the $\hat{p}_\alpha - w_\alpha$ relationship and incorporating this relationship in the discrete kinematic constraint allows us to directly control the nature of the constraint enforcement for a given overlapping pair. Effectively, we are able to control the enforcement of a zero pressure constraint that allows for separation of contacting face pairs, and a zero gap constraint, which results in a positive contact pressure solution. Doing so allows us to robustly handle an active set with face pairs that may come out of contact through an increment of deformation. The design and algorithmic implementation of $(\alpha_\alpha, \beta_\alpha)$ selection is explained thoroughly in Section 10 and the efficacy of this method is demonstrated in Section 11. An important note about the relationship in Equation 9.70 is that it assumes each pair of $\hat{p}_\alpha - w_\alpha$ unknowns is decoupled from the rest. As a note, this is not the case as each finite element node that belongs to a face that in turn is used to construct an ω_α overlap is implicitly dependent on other nodes. Nevertheless, treating Equation 9.70 as a decoupled relationship is ultimately handled by a subcycling procedure fully outlined in Section 10.

9.4.9 Notes on the Linearization of the Kinematic Constraint and the Global Tangent Stiffness

The total derivative of Equation 9.73 is written as

$$D(\beta_\alpha g_\alpha - \alpha_\alpha \hat{p}_\alpha) = \beta_\alpha \left(\frac{\partial g_\alpha}{\partial \hat{u}_k} + \frac{\partial g_\alpha}{\partial \hat{p}_\beta} \right) - \alpha_\alpha \left(\frac{\partial \hat{p}_\alpha}{\partial \hat{u}_k} - \frac{\partial \hat{p}_\alpha}{\partial \hat{p}_\beta} \right), \tag{9.76}$$

where

$$\frac{\partial g_\alpha}{\partial \hat{u}_k} = \frac{\partial \bar{n}_{i\alpha}}{\partial \hat{u}_k} \cdot g_{i\alpha} + \bar{n}_{i\alpha} \cdot \frac{\partial g_{i\alpha}}{\partial \hat{u}_k}, \tag{9.77}$$

$$\frac{\partial g_\alpha}{\partial \hat{p}_\beta} = 0, \tag{9.78}$$

$$\frac{\partial \hat{p}_\alpha}{\partial \hat{u}_k} = 0, \tag{9.79}$$

and

$$\frac{\partial \hat{p}_\alpha}{\partial \hat{p}_\beta} = \delta_{\alpha\beta}. \tag{9.80}$$

Focusing on Equation 9.77, we established in Section 9.3.3 that ω_α is fixed for a Newton iteration. Therefore, $\frac{\partial \bar{n}_{i\alpha}}{\partial \hat{u}_k} = 0$. Furthermore, Equation 9.78 results from the fact that Equation 9.66 does not depend on the unknown contact pressure coefficients. Therefore, Equation 9.76 reduces to

$$D(\beta_\alpha g_\alpha - \alpha_\alpha \hat{p}_\alpha) = \beta_\alpha \left(\bar{n}_{i\alpha} \frac{\partial g_{i\alpha}}{\partial \hat{u}_k} \right) - \alpha_\alpha \delta_{\alpha\beta}. \tag{9.81}$$

For the case where the subscript $\beta = \alpha$, we have

$$D(\beta_\alpha g_\alpha - \alpha_\alpha \hat{p}_\alpha) = \beta_\alpha \left(\bar{n}_{i\alpha} \frac{\partial g_{i\alpha}}{\partial \hat{u}_k} \right) - \alpha_\alpha, \tag{9.82}$$

where no sum on α is implied. Focusing on the first term of Equation 9.82, let the current configuration nodal coordinates in Equation 9.65 be $\mathbf{x}_a^{(i)} = \mathbf{X}_a^{(i)} + \mathbf{U}_a^{(i)} + \hat{\mathbf{u}}_a^{(i)}$, where $\mathbf{X}_a^{(i)}$ is the reference configuration position, $\mathbf{U}_a^{(i)}$ is the beginning time-step displacement, and is $\hat{\mathbf{u}}_a^{(i)}$ is the incremental nodal displacement at the current Newton iteration, all of which are for

node a on side i. As a result, we may write Equation 9.65 as

$$\mathbf{g}_\alpha(\bar{\mathbf{x}}) = \frac{1}{|\omega_\alpha|} \int_{\omega_\alpha} \Bigg(\sum_a N_a^{(1)}(\xi(\bar{\mathcal{P}}_\alpha^{(1)}(\bar{\mathbf{x}})))(\mathbf{X}_a^{(1)} + \mathbf{U}_a^{(1)} + \hat{\mathbf{u}}_a^{(1)}) - \sum_a N_b^{(2)}(\xi(\bar{\mathcal{P}}_\alpha^{(2)}(\bar{\mathbf{x}})))(\mathbf{X}_b^{(2)} + \mathbf{U}_b^{(2)} + \hat{\mathbf{u}}_b^{(2)}) \Bigg) \mathrm{da} = 0, \quad \forall\, \bar{\mathbf{x}} \in\ \omega_\alpha. \tag{9.83}$$

Therefore, the derivative of the first term on the right hand side of Equation 9.82, which is with respect to incremental nodal displacements, is

$$\bar{n}_{i\alpha} \frac{\partial g_{i\alpha}}{\partial \hat{u}_{jd}} = \bar{n}_{i\alpha} \int_{\omega_\alpha} N_d^{(i)} \mathrm{da}, \tag{9.84}$$

which is a column of $\hat{\mathbf{M}}$ associated with ω_α and node c on side i. Consequently, the tangent stiffness contribution from the gap equation at a single overlap is,

$$\beta_\alpha \bar{n}_{i\alpha} \frac{\partial g_{i\alpha}}{\partial \hat{u}_{ja}} = \beta_\alpha [\hat{m}]_{jai\alpha}, \text{no sum on } \alpha, \tag{9.85}$$

where $\hat{\mathbf{M}}^{\mathrm{T}} = [\hat{m}]_{jai\alpha}$.

9.5 Numerical Integration of Contact Integrals

It has been shown that the elements of $\hat{\mathbf{M}}$ are the contact integral contributions to the contact portion of the finite element residual, $\hat{\mathbf{R}}$. $\hat{\mathbf{M}}$ is also the block tangent stiffness matrix in the global system of equations comprised of contributions from the $\hat{\mathbf{R}}$ derivative terms, while $\hat{\mathbf{M}}^T$ is the block tangent stiffness matrix comprised of contributions from derivatives of the weak form of the discrete gap equations. Consequently, the contact specific integral evaluations are those in $\mathbf{M}$ (whose columns are used to form $\hat{\mathbf{M}}$), and the gap equation, Equation 9.63. To perform these integral evaluations in 3D, two different numerical integration schemes are

used. The discretization of the contact integrals into integrals over a collection of discrete intermediate surfaces requires some careful discussion.

The discretized contact interface is such that $\gamma_c^{(1)} = \gamma_c^{(2)} \approx \gamma_c = \cup_{\alpha=1}^{N} \omega_\alpha$, N being the total number of local overlaps, each associated with a different p_α constructed from a local face-face interaction. With this in mind, the intermediate surfaces, ω_α, are the surfaces over which contact integrals are evaluated. This notion is somewhat abstract since γ_c, as defined, is neither $\gamma^{(1)}$ nor $\gamma^{(2)}$, which appears contrary to the idea of a boundary in a boundary value problem. To ease the abstraction, another way to look at the contact interface is by projecting each ω_α back onto its associated $f_j^{(1)}$ and $f_k^{(2)}$ faces. These projections are the image of ω_α projected onto $\gamma_c^{(1)}$ and $\gamma_c^{(2)}$ in the direction of $\bar{\mathbf{n}}_\alpha$ and are denoted as $\omega_\alpha^{(1)}$ and $\omega_\alpha^{(2)}$. In this way we may think of the discretized current configuration contact boundaries as $\gamma_c^{(1)} = \cup_{\alpha=1}^{N} \omega_\alpha^{(1)}$ and $\gamma_c^{(2)} = \cup_{\alpha=1}^{N} \omega_\alpha^{(2)}$, which are composed of separate and distinct collections of $\omega_\alpha^{(i)}$, but are linked through a *common* collection of intermediate surfaces, ω_α.

An important note when considering the latter paradigm, however, is that $|\omega_\alpha| \neq |\omega_\alpha^{(1)}| \neq |\omega_\alpha^{(2)}|$; consequently, equal-and-opposite contact nodal forces is lost. To avoid this, one must include an area scaling in the integrals over $\omega_\alpha^{(i)}$. The paradigm, and algorithmic choice, for that matter, is largely a matter of taste. In the first, a collection of intermediate contact surfaces *common* to both $\gamma_c^{(1)}$ and $\gamma_c^{(2)}$ is used, directly forming a common γ_c over which to enforce contact. The latter preserves the use of the original current configuration contact boundaries of each surface, which are the exact boundaries present in the weak form contact integrals, but requires an area scaling to preserve equal-and-opposite contact nodal forces. The key consideration in both paradigms is that there is a unique mapping of a point on a contact facet in the current configuration to the intermediate surface, or median plane domain, with which it is associated. There is *also* a unique mapping defined from a point on an intermediate surface, ω_α, to each facet that is used to form ω_α. This mapping provides a congruency between both boundary paradigms. This work performs all integration on ω_α,

out of algorithmic considerations.

Let us recall the nonzero integral components of $\mathbf{M}$ shown in Equations 9.29- 9.31 where a_α is taken to be 1:

$$m_{0a}^{(i)\alpha} = \bar{\mathbf{n}}_\alpha \int_{\omega_\alpha} \bar{N}_a^{(i)} da, \tag{9.86}$$

$$m_{1a}^{(i)\alpha} = \bar{\mathbf{n}}_\alpha \int_{\omega_\alpha} \bar{N}_a^{(i)} x da, \tag{9.87}$$

and

$$m_{2a}^{(i)\alpha} = \bar{\mathbf{n}}_\alpha \int_{\omega_\alpha} \bar{N}_a^{(i)} y da, \tag{9.88}$$

where x and y are linear monomial terms that parameterize the 2D intermediate surface, ω_α, and $\bar{N}_a^{(i)}$ are the finite element shape functions defined on the face $\bar{f}_j^{(i)}$ associated with the intermediate surface, ω_α defined by unit normal, $\bar{\mathbf{n}}_\alpha$, for all nodes $a \in \mathcal{A}_\alpha^{(i)}$. A given integration point position on ω_α is then inversely mapped to the face's parent space to obtain a $\boldsymbol{\xi}$ position, at which an $\bar{N}_a^{(i)}$ function evaluation may be made performed. One numerical integration method is used to evaluate the integrals in Equations 9.86- 9.88.

Conversely, the gap equation, Equation 9.63, was derived using the isoparametric mapping of the current configuration faces for each face-face interaction. Specifically, a given integration point on ω_α will be projected back to an associated face, $f_j^{(i)}$, and that position vector on the current configuration face will be inversely mapped to the face's parent element to obtain a $\boldsymbol{\xi}$, for which an $N_a^{(i)}(\boldsymbol{\xi})$ function evaluation may be obtained. As a result, we cannot use $\bar{N}_a^{(i)}$ for the integrands in Equation 9.63, as the inverse isoparametric mapping is that for the current configuration face, $f_j^{(i)}$. A separate numerical integration method is used to evaluate the integrals in Equation 9.63.

9.5.1 Quadrature on an Arbitrary Polygon using a Polynomial Fitting Scheme

The **M**-integrals are computed using a quadrature rule based on a polynomial fitting scheme presented in [25]. Recall the general quadrature form,

$$\int_{\omega_\alpha} f(x_1, x_2)da = \sum_{i=1}^{nip} w_i f_i,\ f_i = f(x_1^i, x_2^i), \tag{9.89}$$

where nip is the number of integration points and w_i are the integration point weights. Let us then define a quadratic polynomial fitting function

$$\tilde{f}(x_1, x_2) = a_0 + a_1 x_1 + a_2 x_2 + a_3 x_1^2 + a_4 x_1 x_2 + a_5 x_2^2. \tag{9.90}$$

Given function evaluations, $f(x_1, x_2)$, at the polygon's vertices and edge mid-points, (i.e. $2n$ chosen integration points on $\partial\omega_\alpha$, where n is the number of vertices on the polygon, ω_α), we solve the minimization problem

$$\min_{a_i} [\tilde{f}(x_1^i, x_2^i) - f_i]^2 p_i, \tag{9.91}$$

where p_i is the least squares weight at the i^{th} discrete data (i.e. integration) point, which is assumed to be known and will be discussed shortly.

The following is a representation of all of the discrete quadratic polynomial evaluations at the integration points:

$$\mathbf{Xa} = \tilde{\mathbf{f}}. \tag{9.92}$$

Equation 9.92 is, in expanded form,

$$\begin{bmatrix} 1 & x_i & y_i & x_i^2 & x_i y_i & y_i^2 \\ 1 & x_{i+1} & y_{i+1} & x_{i+1}^2 & x_{i+1}y_{i+1} & y_{i+1}^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{nip} & y_{nip} & x_{nip}^2 & x_{nip}y_{nip} & y_{nip}^2 \end{bmatrix} \begin{Bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{Bmatrix} = \begin{Bmatrix} \tilde{f}_1 \\ \tilde{f}_2 \\ \vdots \\ \vdots \\ \vdots \\ \tilde{f}_{nip}, \end{Bmatrix} \tag{9.93}$$

where $(x_1^i, x_2^i) \rightarrow (x_i, y_i)$ for economy of notation. The minimization of Equation 9.91 leads to the following set of weighted least squares equations:

$$\mathbf{X}^{\mathrm{T}}\mathbf{W}\mathbf{X}\mathbf{a} = \mathbf{X}^{\mathrm{T}}\mathbf{W}\mathbf{f}, \tag{9.94}$$

where $\mathbf{W}$ is a diagonal matrix of weights p_i (to be discussed shortly), and $\mathbf{f}$ is the vector of actual integrand evaluations at the integration points. To continue, we define

$$\mathbf{A} = \mathbf{X}^{\mathrm{T}}\mathbf{W}\mathbf{X}, \tag{9.95}$$

$$\mathbf{B} = \mathbf{X}^{\mathrm{T}}\mathbf{W}, \tag{9.96}$$

and

$$\mathbf{a} = \mathbf{A}^{-1}\mathbf{B}\mathbf{f}. \tag{9.97}$$

Next, substitute $\tilde{f}(x_1, x_2)$ into the left hand side of Equation 9.89, yielding

$$\begin{aligned}\int_{\omega_\alpha} \tilde{f}(x_1, x_2) =& a_0 \int_{\omega_\alpha} da + a_1 \int_{\omega_\alpha} x_1 da + a_2 \int_{\omega_\alpha} x_2 da + \\ & a_3 \int_{\omega_\alpha} x_1^2 da + a_4 \int_{\omega_\alpha} x_1 x_2 da + a_5 \int_{\omega_\alpha} x_2^2 da,\end{aligned} \tag{9.98}$$

which may be written as

$$\int_{\omega_\alpha} \tilde{f}(x_1, x_2) = \mathbf{h}^{\mathrm{T}} \mathbf{a}, \tag{9.99}$$

where $\mathbf{h}^{\mathrm{T}}$ is a vector containing the integrals in Equation 9.98, and $\mathbf{a}$ is the vector of known coefficients of the quadratic polynomial fitting function. The evaluation of the integrals in Equation 9.98 is performed using the divergence theorem and evaluating line integrals on the boundary of ω_α. Since we have an expression for $\mathbf{a}$, we may write Equation 9.99 as

$$\int_{\omega_\alpha} \tilde{f}(x_1, x_2) = \mathbf{h}^{\mathrm{T}} \mathbf{A}^{-1} \mathbf{B} \mathbf{f}. \tag{9.100}$$

Pluggin Equation 9.100 into Equation 9.89 allows one to solve for the unknown quadrature weights, the solution of which is expressed as

$$\mathbf{w}^{\mathrm{T}} = \mathbf{h}^{\mathrm{T}} \mathbf{A}^{-1} \mathbf{B}. \tag{9.101}$$

Equation 9.101 provides the integration point weights on the right hand side of Equation 9.89, and thus, completes the quadrature rule using a quadratic polynomial fitting function with discrete function evaluations at the integration points on the boundary of the polygon. This numerical integration method is well-suited for the evaluations of the **M**-integrals using a convenient 2D parameterization of the projected facets, $\bar{f}_j^{(i)}$, and associated intermediate surface, ω_α, based on the median plane, p_α.

9.5.2 Quadrature on an Arbitrary Polygon using Triangular Partitioning

The gap equation, Equation 9.63, involves integrals of shape functions associated with current configuration faces, integrated over intermediate surfaces, ω_α. The projection of a point, $\bar{\mathbf{x}} \in \omega_\alpha$, onto the current configuration face, $f_j^{(i)}$, in the direction of $\bar{\mathbf{n}}_\alpha$, is performed in the global coordinate system. Rather than having a global position vector *and* a 2D parameterization of that position vector in the plane, p_α, in order to use the polynomial fitting scheme to derive a quadrature rule on ω_α, it proved convenient to handle the numerical integration of Equation 9.63 in 3D, current configuration coordinates.

A quadrature rule on triangles, outlined in [26], and called the TWB method, was used to integrate Equation 9.63 in 3D current configuration coordinates. The intermediate surface, ω_α, will generally be a convex polygon and may be subdivided into triangles using the vertices of the polygon and the vertex averaged centroid. Using barycentric coordinates, [26] provides precomputed integration points and weights for the exact integration of successively higher order polynomials. In this work,three integration points per triangle were used to integrate Equation 9.63. There is one triangle per edge in a polygon, so there are $3n$ integration points where n is the number of edges (equal to the number of vertices) of the polygon.

To compare the adequacy of this method to that of the polynomial fitting, we require that

$$\sum_a \int_{\omega_\alpha} N_a da = |\omega_\alpha|. \tag{9.102}$$

Computation of Equation 9.102 using the polynomial fitting scheme and the TWB method reproduce the area of the intermediate surface to working precision. The choice of which method to use has been explained to be a matter of algorithmic convenience given the nature of the required integral computations.

9.5.3 Inverse Isoparametric Mapping

The inverse isoparametric mapping required for the numerical integration of the contact integrals, per the methods outlined in the previous two sections, is explained in this section. The general form of the isoparametric mapping is

$$\mathbf{x} = \sum_{a}^{nnd} N_a(\xi, \eta)\mathbf{x}_a, \tag{9.103}$$

where nnd is the number of nodes for a particular face and N_a are the shape functions associated with that face, where $\mathbf{x}_a$ are the nodal coorindates in some physical configuration. The shape functions, which are the functions to be evaluated in Equation 9.103, are defined in (ξ, η)-space, or parent space for two-dimensional facet elements. The issue is that we have an integration point in physical space and need to determine its corresponding point in (ξ, η)-space. What this means is that $\mathbf{x}$ on the left hand side of Equation 9.103 is known, as are the physical configuration nodal coordinates, but (ξ, η) with which to evaluate N_a is not known. To solve this problem we write the following equation

$$\mathbf{f}(\xi, \eta) = \mathbf{x} - \sum_{a}^{nnd} N_a(\xi, \eta)\mathbf{x}_a = 0. \tag{9.104}$$

We solve Equation 9.104 using Newton's method, which requires us to form the Jacobian of $f(\xi, \eta)$, which is based on the specific shape functions for the facet element in question. A note to be made here is that the Jacobian, or the total derivative of $\mathbf{f}(\xi, \eta)$, $\mathbf{Df}(\xi, \eta) = \mathbb{R}^{3x2}$, which is not a square matrix. That is, if we use the three dimensional nodal coordinates, $\mathbf{x}_a$, with a two dimensional function $N_a(\xi, \eta)$, for which (ξ, η) are unknowns, then the Jacobian formed in Newton's method is not square. If we denote the system of equations to be solved in each Newton's iteration generically as, $\mathbf{Ax} = \mathbf{b}$, where for clarity $\mathbf{A}$ represents the Jacobian, $\mathbf{Df}(\xi, \eta)$, $\mathbf{x}$ represents the correction to the vector of unknowns, and $\mathbf{b}$ represents the vector valued function evaluation $\mathbf{f}(\xi, \eta)$, then to create a square system, we simply

premultiply each side by $\mathbf{A}^{\mathrm{T}}$. This gives us the system of equations, $\mathbf{A}^{\mathrm{T}}\mathbf{A}\mathbf{x} = \mathbf{A}^{\mathrm{T}}\mathbf{b}$. Given that $\mathbf{A}^{\mathrm{T}}\mathbf{A}$ is square and invertible, we may solve this system in order to obtain the update to the solution. Since the (ξ, η) parent space coordinate system is centered at (0,0) for a four node quadrilateral, which is at the centroid of the element, this coordinate was chosen as an initial guess to solve for the (ξ, η) pair that correspond to a given position vector, $\mathbf{x}$.

9.6 Global Equations

Using the notes on the tangent stiffness contributions found in Sections 9.3.3 and 9.4.9, we may write the global system of equations for a given Newton iteration as

$$\left[\begin{array}{c|c} \bar{\mathbf{K}} & \hat{\mathbf{M}} \\ \hline \mathbf{B}\hat{\mathbf{M}}^{\mathrm{T}} & \mathbf{A} \end{array}\right] \left\{\begin{array}{c} \delta\hat{\mathbf{u}} \\ \hline \hat{\mathbf{p}} \end{array}\right\} = \left\{\begin{array}{c} \mathbf{R} \\ \hline \mathbf{c} \end{array}\right\}, \tag{9.105}$$

where $\bar{\mathbf{K}}$ is the stiffness contribution from the $\bar{\mathbf{R}}$ term in the residual, and $\hat{\mathbf{M}}$ is the stiffness contribution from the $\hat{\mathbf{R}}$ term. The bottom left off-diagonal block, $\mathbf{B}\hat{\mathbf{M}}$, is the stiffness block derived from the gap equation of the equality constraint equations. The matrix $\mathbf{B}$ is a diagonal matrix with the β_α parameters associated with the inequality constraint as the diagonal entries. Furthermore, the bottom diagonal block, $\mathbf{A}$, is a diagonal matrix with the α_α parameters associated with the slack variable term in the inequality constraint. The stacked vector of unknowns contains the corrections to the incremental nodal displacements, $\delta\hat{\mathbf{u}}$, and the unknown contact pressure coefficients, $\hat{\mathbf{p}}$. The right hand side of Equation 9.105 is composed of the equilibrium residual evaluation, $\mathbf{R}$, at the current iterate, and the associated vector of constraints, $\mathbf{c}$, at each ω_α overlap.

An unconventional aspect of this system of equations is that the size of the system, or rather, the size of $\hat{\mathbf{M}}, \hat{\mathbf{p}}$, and $\mathbf{c}$ change from one Newton iteration to the next as faces come in and out of contact. An important algorithmic consideration in the implicit solution setting is that

while one is looping over Newton iterations, the incremental nodal displacement solution is being updated with each subsequent iterate's corrections. Each correction may result in nodal displacements that take current contact faces out of contact or previously contact-free faces into contact. To handle this adequately, one must update the contact interface, i.e. the collection of ω_α face overlaps, each Newton iteration. A caveat to this point is that the appropriate constraint to enforce at each overlap, that of zero pressure or zero gap, is not known apriori. As a result, the business of choosing the correct setting of each $(\alpha_\alpha, \beta_\beta)$ parameter pair for each overlap will be achieved via a subcycling procedure occuring on a Newton iteration per Newton iteration basis. This subcycling solution procedure is explained in detail in Section 10.

10 Numerical Implementation and Solution Procedure

The implementation of the contact methodology was performed in `Imitor`, a finite element research code for the solution to problems in nonlinear solid mechanics. The implementation was done in three dimensions with flexible data structures to accomodate a future two dimensional implementation. While the focus of this work is on quasi-static nonlinear solid mechanics, `Imitor` has an implicit dynamics implementation. Extending contact to include dynamics, then, is a straightforward task. This work focuses on a quasi-static implemenation because of the belief that the primary difficulty in implicit contact lies in quasi-static analysis with dynamics simply being an extension of a robust algorithm developed under the former analysis type. While the purpose of this work is not to exhaustively detail the structure of `Imitor` itself, some explanation will prove useful when discussing the contact implementation and associated data structures.

The following section introduces `Imitor` and discusses the code at a high level. The subsequent sections present the algorithmic details and pertinent data structures for the contact implementation, which can be separated into four parts: a *geometric* part including contact search and computation of contact element data, a part defining contact *elements* and contact *nodes*, a *solution pass* part including subcycling and convergence criteria, and the *solution procedure* specific to this contact implementation.

10.1 Introduction to `Imitor`

`Imitor` is a finite element research code written in modern Fortran using the 2008 standard ([27]). The code supports geometric and material nonlinearity and is written in a flexible manner to handle an arbitrary number of element formulations, physics types, and solution procedures. The code is built off a derived type called `SIM_MESH`. This data structure holds all

pertinent analysis data such as convergence parameters, time step data, boundary conditions, material data, solver data, and element and node data, and connectivity (i.e. mesh data). The overarching idea behind the `SIM_MESH` data structure is that when an analysis begins, the entire populated `SIM_MESH` object *advances*, so to speak, from one timestep to the next through a series of *solution passes*. In the most general setting, a solution pass is either a Newton step involving a solve, or a convergence check involving a residual evaluation. At any point in the analysis, the `SIM_MESH` object holds the data of the most current state of that analysis. Populate such an object, pass it to the main `Imitor` program, and you will get a solution at each time step in return. Given the importance of this data structure, the contact implementation is extended off the `SIM_MESH` derived type.

10.1.1 The Contact Interaction

While much of the contact data is stored as component objects on the `SIM_MESH` derived type, the contact implementation is primarily built off a component object on `SIM_MESH` of derived type `CONTACT_INTERACTION`. A *contact interaction* is defined by the finite element faces belonging to two possibly contacting/interacting surfaces.[4] This is a two surface implementation, which is to say that a contact interaction is defined by user prescribed collections of facets on a user defined surface 1 and surface 2. A *surface* need not be a *contiguous* collection of facets on the boundary of a body, and need not even be defined by a collection of facets on a *single* body. For example, we may define a contact interaction between a collection of facets on the boundary of body 1, which defines surface 1, and two separate collections of facets on the boundry of body 2 and body 3, which defines surface 2. That is, we have defined an interaction whereby a portion of the boundary of body 1 may come into contact with a portion of the boundaries of bodies 2 and 3. The important detail when

[4] In three dimensions we may refer to the faces as two dimensional facets, and in two dimensions we may refer to the faces as one dimensional segments, which agrees with typical nomenclature. This work focused on the three dimensional implementation, and so faces will often be referred to specifically as facets.

defining a contact interaction is that no facet from one interaction may be included in an altogether separate interaction. For instance, using the aforementioned interaction example, the prescribed collections of facets on the boundary of body 2 and body 3 cannot form an additional contact interaction. If this were desired, the current formulation and numerical methodology would have to extend itself to three-body or more contact interactions. This does, in principle, create a limitation on the complexity of the contact interactions between multiple bodies, but in practice, does not pose a significant restriction on the user's ability to adequately prescribe contact interactions for a desired simulation. An additional note is that self-contact is accounted for where in the simplest setting one may prescribe two separate, non-intersecting, contiguous collections of facets on the same boundary of a single body to define a contact interaction.[5]

Given a user prescribed interaction, the `CONTACT_INTERACTION` object on the `SIM_MESH` object contains geometric data for every facet in that interaction. This data is stored in an array of `CONTACT_FACET` objects, which is a component of the `CONTACT_DAT` derived type. Each `CONTACT_INTERACTION` object has a component object of derived type, `CONTACT_DAT`, holding the facet geometric data.

The main type bound procedure (TBP) acting on a `CONTACT_INTERACTION` object takes the facet data defining an interaction and performs the geometric search to determine which facets constitute a contact pair. The data of each pair is then stored in a binary search tree on the `CONTACT_INTERACTION` object, to later populate contact element and contact node binary search trees on the `SIM_MESH` object. The next section describes the contact search.

[5]Non-intersecting in this sense (i.e. self-contact) means that no polytope belonging to any facet on one prescribed surface of the interaction also belongs to a facet on the other prescribed surface.

10.2 Contact Geometry

The contact geometry, in this instance, refers to the contact search and element formulation. What is meant by the contact search is the search for face-on-face pairs that violate, or *nearly* violate, the kinematic constraint. Once such face-pairs have been identified, the contact element is principally defined by its intermediate surface, which is the intersection, or overlap, between the two faces when projected onto the median plane. The intermediate surface is the surface over which we compute the contact integrals. The following sections outline these two aspects of the contact algorithm in more detail.

10.2.1 Contact Search

The contact search is typically a computationally expensive procedure within a contact implementation and is a fairly mature subject in the literature. While the search was not the focus of this work, it is worth a cursory discussion pointing the reader in the direction of some rather excellent resources on the matter. The books [10], [6] and [28] provide an excellent overview of the contact search. Specifically, Wriggers (see [10]) separates the contact search into two parts: the spatial or geometric search and contact detection. The first has to do with breaking down associated contact geometry in such a way that one can efficiently carry out the contact detection search. A straightforward implementation of this is to form bounding boxes around each possibly contacting interface and only perform a contact detection search for facet-pairs whose bounding boxes touch. While there are some obvious details omitted for simplicity, the point of this search is the geometric association of portions of opposing contact surfaces. The contact detection is just as the name implies, the search for facet-pairs that are actually deemed to be in contact. Examples including bucket sort, heap sort and octree methods are found in [8], [29] and [30]. Furthermore, the work found in [31] and [32] treat the contact search in the context of mortar methods.

In this work, a simplified approach was taken. This approach is described as follows. For a given contact interaction, the radius of each facet's enclosing sphere is computed. Then, every opposing facet-pair whose centroids are separated by a distance (i.e. vector magnitude) of no greater than the sum of the two facets' enclosing spheres' radii are considered *proximate*. This calculation is performed n^2 times where n is the total number of facets in an interaction; however, the calculation simply involves low level floating point operations. This calculation, herein called the proximity search, is used to narrow down facet-pairs with which to proceed with more complex geometrix calculations.

For *proximate* facet-pairs, the next level in the geometric search is to compute the maximal or minimal distance of interpenetration or separation per the method outlined in Section 9.4.2. This step in the search determines which *proximate* pairs are actual *contact* pairs. For facets exhibiting interpenetration, we obviously have a contact interaction and contact pair. As described in Section 9.4.3, however, we may be interested in defining a contact pair between facets that exhibit separation up to some tolerance. As discussed in Section 9.4.2, the collision detection part of the geometric search provides a median plane. For those facets that exhibit separation up to some tolerance, one additional check is to make sure that the facets projected onto the median plane intersect. That is, there must be nonzero projected area of overlap for these facet-pairs. Additionally, for separation *and* interpenetration, the magnitude of the area of overlap of the projected facets is checked against some tolerance. Specifically, the following criteria holds for the magnitude of this area,

$$\frac{|\omega_\alpha|}{min(|\bar{f}_j^{(1)}|, |\bar{f}_k^{(2)}|)} > tol. \tag{10.1}$$

The denominator in Equation 10.1 is the smaller of the two projected facet areas that are used to construct the overlap area, ω_α. The reason behind this can be explained as follows. Imagine one "large" facet and one "small" facet in a contact interaction. As the overlap area approaches the size of the small facet, the contact overlap becomes more significant in

resolving the contact interaction over the side to which the smaller facet belongs. While the contact area may not play a significant role on the side to which the larger facet belongs, if we were to use a *max* area in the denominator in Equation 10.1, we may, in principle, lose contact interactions that ought to play a role in enforcing contact behavior on the side with the smaller element size.

The geometric search contains an outer loop over facets on surface 1 of the contact interaction and an inner loop over facets on surface 2 of the contact interaction. The inefficiency of this search lies in the $\mathcal{O}(n^2)$ proximity check. This is an area where modern search methods will prove useful in enhancing contact search efficiency. The subroutines that carry out the contact search are type bound procedures on the `CONTACT_DAT` component object on the `CONTACT_INTERACTION` object. The `CONTACT_DAT` object contains a component object array of derived type `CONTACT_FACET`. It is from this array of facet objects that the data for a given facet-pair is aggregated and used to populate a local contact data object on the `CONTACT_DAT` object. The local data object is an object of derived type `DATA_OBJ_FCT_FCT_DAT`, that temporarily holds facet-pair data in order to perform geometric calculations. If, based on the collision detection data and projected area tolerance check, a facet-pair is deemed to be in contact, then further calculations associated with the contact element formulation are performed on the local contact data object.

10.2.2 Contact Element Data

Before we discuss the contact element itself, we must discuss the element data, which one may consider the backbone behind a contact element formulation. The element data is built on the definition of the median plane constructed from a facet-pair in the current configuration. The median plane data is stored on the local contact data object on the `CONTACT_DAT` component object on the `CONTACT_INTERACTION` object of `SIM_MESH`. The median plane is determined

per Section 9.4.2. Once a median plane is established, each facet is projected onto it along the direction of the plane's normal. The intersection of the projected facets forms the intermediate surface over which contact integrals are computed for a given facet-pair. This intermediate surface is the area of overlap, ω_α. The data used to construct and define the median plane and intermediate surface is purely geometric data stored on a local data object.

What makes one contact element fundamentally different from another contact element is the number of unknown pressure coefficients that are "active" in the pressure basis. That is, when we compute the $\mathbf{M}$ operator per Section 9.2 and modify $\mathbf{M}$ by constructing a full column-rank operator, $\hat{\mathbf{M}}$, per Section 9.3, we determine the active pressure basis coefficients for each contact element. As a result, we may consider a contact element formulation to be defined by the number of active coefficients with associated columns in $\hat{\mathbf{M}}$ that contribute to the pressure basis. Before we can construct $\hat{\mathbf{M}}$, however, we must compute all $\mathbf{M}$ integrals for each facet-pair. This is the contact integral data that is stored with the geometric data on the local contact data object. The integral computations are performed on a pair-by-pair basis and are carried out over a given pair's intermediate surface. This data is stored on each instance of a local contact data object, which again stores the median plane data, intermediate surface data, and now, the $\mathbf{M}$ integrals for the contact pair at hand in the search algorithm. In this implementation, we use a constant pressure basis. As a result, there is no reason to construct a full column-rank operator per Section 9.3 since a constant pressure basis is guaranteed to produce a full column-rank operator. As a result, $\hat{\mathbf{M}} = \mathbf{M}$. Furthermore, this means that each contact element formulation is the same. If one were to desire a richer pressure basis, the construction of $\hat{\mathbf{M}}$ would occur after the contact search and associated element formulation calculations.

At this point a local contact data object has been populated with the median plane data, the intermediate surface data, and the $\mathbf{M}$ integrals specific to each facet-pair's intermediate surface geometry. Once the requisite data has been computed, the local data object is used to

populate a binary search tree on the CONTACT_INTERACTION object. The local contact object is then deleted, and the calculations and population of a new local contact data object is performed for the next candidate facet-pair. A binary search tree is used to store the contact pair data for its ability to handle an arbitrary number of data "nodes", which is important since we do not a priori know the number of contacting facet-pairs at any increment of deformation in the simulation. Additionally, the efficient and unordered "lookup" features of a binary search tree were attractive when first designing the contact implementation. While traversing a binary search tree is preferred, it was not immediately clear if a more robust lookup capability was needed. This feature proved useful in a few instances.

While we have spoken of a contact element formulation, the local contact data object and the populated binary search tree simply hold contact facet-pair *data*. As stated, this includes geometric data and contact integral evaluations. The actual contact elements are formed once the binary search tree is full of all contact pair data, and the element objects are populated only with the data necessary to compute residual and/or tangent stiffness contributions. Once all contact facet-pair data objects have been inserted into the binary search tree, then the CONTACT_INTERACTION object is considered fully populated. The CONTACT_INTERACTION data is then used to create contact element and contact node data structures, which are stored on the SIM_MESH object. The following two sections outline this portion of the contact implementation.

10.3 Formation of Contact Elements and Nodes

The contact data stored in the binary search tree on the CONTACT_INTERACTION component object array on SIM_MESH is used to populate contact elements and nodes, which are stored in their own respective binary search trees on SIM_MESH. This aggregation of data across all contact interactions into element and node objects allows for a natural inclusion of contact

elements into the element assembly process and contact nodes into convergence subroutines, for example, where loops over nodes are used.

It must be noted that at the time of the initial design of the contact implementation, it was not immediately clear how the contact elements/nodes would be populated within the solution procedure. As a result, the most flexible dynamic data structure at hand was in fact the binary search tree. The contact implementation has undergone various permutations, which now makes it clear that since each contact interaction holds a complete binary search tree of contact data objects, we could use allocatable array component objects on `SIM_MESH` to store the contact elements and nodes, because at this point we do know the total number of interacting facet-pairs. This would be permissible since anytime new contact pairs are required during the solution procedure, the contact data pairs are built from the ground up without regard to previous facet interactions. Clearly, one can imagine that this is not the most optimal way to form/reform contact facet-pairs, but it was the most logical and straightforward to implement. Using binary search trees to store contact data *and* elements/nodes, however, allows for the addition or removal of data objects to or from the tree, respectively. As a result, a more optimal implementation of contact data/element/node formation utilizing previous contact facet-pair data may be more easily implemented.

10.3.1 Contact Elements

`Imitor` defines an element as any geometric entity over which residual and/or tangent stiffness contributions are computed. As a result, a two dimensional facet over which a Neumann boundary condition is prescribed is treated as a *separate* element from the three dimensional hexahedron volume element to which the facet belongs. Following this structure, each intermediate contact surface is used to form contact elements over which an unknown pressure boundary condition is prescribed. While the number of facet elements or volume elements is

known a priori, which allows for a single loop over "elements" assembling *all* residual and/or tangent stiffness contributions, the number of contact elements is not known prior to a given solution pass. As a result, the data associated with facet and volume elements is contained in an allocatable array, while the data associated with contact elements is contained in a binary search tree. The solution procedure loops over both the conventional element array *and* the binary search tree. Because the number of contact elements changes within a time-step, it was not possible to easily incorporate the contact element data objects in the primary element data array. The secondary binary search tree holds all "dynamic" elements, which offers possible flexibility for new element types in the future (e.g. fracture implementations). While the purpose of this work is not to delve into the data structures of `Imitor` as a whole, the point here is that element assembly is performed by looping over the static element array and the dynamic element array. This process is straightforward enough once an understanding of the element definition is in place and the associated data structures. Furthermore, the addition of new element types very straightforward. As such, we may discuss the notion of a contact element and the calculations that go into its "formulation."

The residual and tangent stiffness contributions from a contact element are centered around the local $\hat{\mathbf{M}}$ integrals (see Sections 9.3.3 and 9.4.9). These integrals are computed and stored on the contact data object. Furthermore, gap calculations will be required during a subcycle convergence check, which requires access to median plane and intermediate surface data (a point that will be discussed in Section 10.4.2). As a result, it proved useful to provide a pointer to the contact facet-pair data as a component on the contact element object. Consequently, the population of contact elements can be viewed as an aggregation of contact data objects, through component pointer objects, under the hood of a single binary search tree. Where the contact element departs from simply having access to contact facet-pair data is that each element contains subcycle parameter data and state variables for that element. In this particular implementation (i.e. frictionless), state variables are not used;

however, subcycle parameters are essential to computing residual and/or tangent stiffness contributions (see Section 9.4.6).

10.3.2 Contact Nodes

The contact node has a geometric location at the centroid of each intermediate surface, though this is somewhat arbitrary. More essential is the data stored on each contact node, which consists of pressure coefficients (i.e. the pressure solution), pressure and gap tolerances for convergence checks, and the scalar slack variable. As such, the node can be thought to store anything related to the pressure degrees of freedom. Since we have expressed the kinematic constraint in an equality form through the use of a slack variable, which itself is related to the unknown pressure at a discrete intermediate surface, the data stored at a contact node is a slightly more complicated affair. In any case, we will need the pressure coefficient solution that is stored on the contact node for contact element residual contributions. This is achieved by creating a *local* element that contains data necessary for residual and/or tangent stiffness calculations. The data on the local element is aggregated as necessary, which means that the pressure coefficient solution on the contact node is retrieved and temporarily stored on the *local* element for the purposes of residual/stiffness calculations. The use of a local element is specific to the data structures of `Imitor`. Additionally, the contact node is used to check convergence. Specifically, the contact node is subject to subcycle convergence criteria, which will be discussed in detail in Section 10.4.3.

The contact search, geometric calculations, and formation of contact elements and nodes is required at various stages throughout the contact solution process. `Imitor` organizes the solution process into various solution *passes* controlled by solution *modes*. Within this pass/mode structure a contact subcycling procedure, which iterates over $(\alpha_\alpha, \beta_\alpha)$ parameters, has been devised and implemented in order to control discrete constraint enforcement. The

subsequent sections present the solution procedure in detail.

10.4 Subcycle Procedure

Section 9.4.7 introduced a linear relationship between the discrete pressure, p_α, and slack variable, w_α, that was introduced in the equality constraint for each overlap ω_α. Section 9.4.4 developed the equality constraint, while Sections 9.4.5 and 9.4.6 discussed the terms in the constraint in detail. The conclusion was made that the exact nature of the constraint enforcement is controlled through the $(\alpha_\alpha, \beta_\alpha)$ parameters, allowing for zero gap contact enforcement or zero pressure separation enforcement. While increasing the robustness of the constraint enforcement for difficult contact problems (something that will be demonstrated in Section 11), there is no way to know, a priori, the exact setting of each $(\alpha_\alpha, \beta_\alpha)$ across the contact interface. As a result, a subcycling procedure has been devised and implemented to iterate toward a converged interface state. The following sections outline this procedure in the context of the implementation in `Imitor`.

10.4.1 Solution Passes

A solution pass is identified by an integer id associated with a solution mode. For the quasi-static, implicit contact implementation performed in this work, the following modes are of interest.

A mode = 1 solution pass is a residual evalution only. The loop over elements simply assembles residual contributions. For contact elements, this amounts to equilibrium residual contributions *and* kinematic constraint evaluations, with global residual contributions from each discrete contact overlap.

A mode = 2 solution pass is a Newton solve. The loop over elements assembles residual *and* tangent stiffness contributions. This presumes an initial setting of all $(\alpha_\alpha, \beta_\alpha)$ parameters for each contact element. At a mode = 2 solution pass, every contact element has an initial α_α setting of 0.5, where β_α is controlled by α_α and a contact element's characteristic stiffness, S_α. These parameters were discussed in detail in Section 9.4. In general, contact will require subcycling over each $(\alpha_\alpha, \beta_\alpha)$ pair following a mode = 2 solve. A key characteristic of this algorithm is that all contact elements are formed prior to a mode = 2 solve[6] and held constant throughout the subcycling. That is, each contact element's intermediate surface, ω_α, is held fixed throughout a Newton iteration + subcycling.

A mode = 4 solution pass is a subcycle convergence check. The first mode = 4 solution pass will follow the first mode = 2 solution pass in a contact analysis. In a regular Newton iteration context, we would normally cycle between mode = 1 residual checks and mode = 2 Newton solves; however, for contact we have introduced a subcycling method. The first mode = 2 solution pass requires an initial setting of the α_α parameters, which are all typically set to 0.5. The Newton solve will provide a pressure solution and corrections to the incremental nodal displacements at each finite element node. The goal of the subsequent mode = 4 solution pass is to check convergence of the pressure-gap solution. That is, for a given setting of α_α, have we effectively enforced a zero gap, positive contact pressure constraint, or a zero pressure, separation constraint? The specifics of this convergence check will be detailed in Section 10.4.3, but the point to be made here is that a mode = 4 check evaluates the enforcement of the kinematic constraints. In general, setting all α_α to 0.5 and computing the associated β_α parameters in the first mode = 2 solution pass will result in too "soft" of a contact enforcement and the first mode = 4 solution pass will not result in convergence. As a result, the first mode = 2 to mode = 4 solution pass sequence will be followed by a mode = 5 subcycle, solve discussed below.

[6] a point that is debated and discussed in Section 10.4.4

A mode = 5 solution pass is a subcycle solve and follows an unconverged mode = 4 subcycle convergence check. A mode = 5 solve is proceeded by another mode = 4 convergence check. The cycling between mode = 4 and mode = 5 constitutes the contact subcycling. Prior to the actual solve itself, a mode = 5 solution pass begins with a *resetting* of the α_α (and corresponding β_α) parameters. While the initial α_α are set to 0.5 to provide some intermediate contact stiffness, the update that takes place in a mode = 5 solution pass resets each α_α to either 0 or 1 (this is done in a procedure that will be outlined in Section 10.4.2). Recall that if $\alpha_\alpha = 1$, then we are enforcing zero pressure with separation, and if $\alpha_\alpha = 0$, then we are enforcing zero gap with positive contact pressure. An intricacy in this subcycling method is that in Section 9.4.7 we outlined a decoupled linear relationship between the pressure and the slack variable (or the negative of the gap), but in reality, each contact element's constituent facets are implicitly coupled to the adjacent facets where there exists multiple contact elements formed by contiguous facets on each surface. The point is that it is not possible, a priori, to intuit the exact setting of every α_α across the contact interface. Rather, while at a given mode = 5 solution pass we set a given $\alpha_\alpha = 1$, but the incremental nodal displacement solution, coupled to the adjacent facets and their contact pressure boundary conditions, may take that contact element back into contact. Two observations are directly drawn from the aforementioned intricacy. One is that the subcycling will generally take more than one mode = 5 to mode = 4 cycle, requiring multiple mode = 5 solves. Two, only through numerical study can we determine whether we get stuck in continuous α_α flipping between 0 and 1. Though the latter is a concern, numerical experimentation using difficult contact problems with converged contact interface states with diverse final settings of α_α parameters has demonstrated that unstable α_α flipping does not occur in practice with this subcycling method. One note regarding this flipping is that the moderate contact stiffness set through the initial α_α setting in a mode = 2 solution pass does a good job of indicating the tendency of a contact element's behavior toward interpenetration or separation and can adequately inform the first mode = 5 α_α update. A related remark regarding this point is

that while the decoupled $p-w$ relationship will lead to multiple subcycles, numerical experimentation has shown that the number of subcycles is often very low. These are qualitative remarks that will be discussed quantitatively in Section 11; however, these are primary observations when first introducing the subcycling method that are worth addressing here to some extent.

10.4.2 Subcycle Update

We have stated that all contact elements are formed prior to a mode = 2 solve. A mode = 2 solve kicks off a Newton iteration, and we have mentioned that it will typically be proceeded by mode = 4/5 subcycling. Let us assume, however, that for a given mode = 2 solve we have set the $(\alpha_\alpha, \beta_\alpha)$ parameters perfectly correctly for the collection of ω_α overlaps. In doing so, we obtain a vector of pressure solutions and corrections to the incremental nodal displacements. The gaps reflected by those corrections pair with the pressure solution at each ω_α to give either zero pressure with separation or positive contact pressure with zero gap; thus, the constraints are adequately satisfied. This unrealistic, ideal case highlights a key progression in thought when considering the more complicated mode = 4/5 subcycling. That is, we subcycle over $(\alpha_\alpha, \beta_\alpha)$ pairs trying to converge to the correct collection of parameter settings to exactly satisfy the constraints. Between subcycles, however, we not only need to actually update the $(\alpha_\alpha, \beta_\alpha)$ pair, but we need to *reset* the incremental nodal displacements of the facets comprising each contact pair back to the original configuration prior to a mode = 2 solve. That is, each subcycle starts with contact pair configurations based on the original construction of each ω_α. Each mode = 5 subcycle solve, then, can be thought of as a solve with an updated mode = 2 system of equations, where each mode = 2 system of equations is the original system of equations based on the current Newton iterate. In a sense this is a predictor-corrector type method where a mode = 5 solve is a prediction based on the setting of all $(\alpha_\alpha, \beta_\alpha)$ pairs. If the prediction does not adequately enforce the constraints, we

use the information from that solve to "correct" the parameter settings and then we reset the incremental nodal displacements in preparation for the next predictor step. The only thing that changes between each "udpated" system is the setting of $(\alpha_\alpha, \beta_\alpha)$ pairs. As such, this update requires some careful discussion keeping the preceeding algorithmic sequence in mind.

The subcycle update is primarily concerned with updating the $(\alpha_\alpha, \beta_\alpha)$ parameters for each ω_α overlap based on a (p_α, w_α) pair. Specifically, the first mode = 5 subcycle solve after a mode = 2 Newton solve produces a vector of pressure solutions as well as a vector of corrections to the incremental nodal displacements. The pressure solution at each ω_α provides the p_α evaluation in the (p_α, w_α) pair; however the w_α evaluation requires some careful discussion. Note that the w_α evaluation is the negative of the gap per Equation 9.66. The slack variable serves to "close" the gap such that the discrete constraints evaluate to zero. As a result, the (p_α, w_α) pair is essentially a pressure-gap evaluation. For simplicity, the slack variable may be referred to as a gap evaluation, since in essence that is what it is. This will make the following discussion a little more intuitive, since for every contact pair, we will refer to a pressure-gap evaluation and corresponding update of the $(\alpha_\alpha, \beta_\alpha)$ pair.

We established that the contact elements' intermediate surfaces are determined prior to a mode = 2 solve and held fixed through subsequent mode = 4/5 subcycling; however, with each mode = 5 subcycle solve, we get an update to the incremental nodal displacements. For facets that comprise a contact element, this update to the incremental nodal displacements reflects the contact stiffness implicit in each $(\alpha_\alpha, \beta_\alpha)$ pair (in addition to any underlying physics in a particular simulation). As a result, in order to gain some insight into how to update the $(\alpha_\alpha, \beta_\alpha)$ pair, we must consider, as previously stated, the pressure solution, but also the updated gap evaluation as well as the prior $(\alpha_\alpha, \beta_\alpha)$ settings. Even though we hold each intermediate surface fixed during the subcycle procedure, which means that the region of integration for the discrete scalar value gap function in Equation 9.66 is held

fixed, we perform an update to the gap evaluation using the update to the incremental nodal displacements. Equation 9.66 is linear in the incremental nodal displacements, making this update trivial. Though we have not updated the intermediate surface based on the corrections to the incremental nodal displacements, we may, nevertheless, perform a gap update. The point worth belaboring a bit here is that there is some error inherent to this approach. The facets comprising a contact element will move as a result of a displacement solution. Since reforming, or rather, updating every contact element's geometric data and contact integrals based on the corrections to the incremental nodal displacements from a subcycle solve may be considered prohibitively expensive, this work allows for some error in the gap update, and handles this at the Newton convergence level. That is, if one component of the convergence criteria for Newton's method involving contact is that the incremental nodal displacements for the nodes belonging to the facets that comprise contact elements is below some tolerance, then the current configuration of intermediate surfaces has effectively converged and the error inherent in the mode = 4/5 gap update using a mode = 2 contact element formulation is negligible. This approach may of course require one or two more Newton iterations, but this cost is considerably less than other alternatives.

An unconverged mode = 4 solution pass, therefore, is equipped with the latest (p_α, w_α) pair, where we have made explicitly clear that the w_α evaluation is a gap update based on a mode = 5 incremental nodal displacement solution using a mode = 2 intermediate surface at each contact element. Using the (p_α, w_α) pair, we may perform an $(\alpha_\alpha, \beta_\alpha)$ update, and reset the incremental nodal displacements to their current, mode = 2 configuration, and perform another mode = 5 solve with the updated mode = 2 system of equations.

The actual update of the $(\alpha_\alpha, \beta_\alpha)$ parameters is informed by Figure 22, which shows a $p_\alpha - w_\alpha$ plot with the four quadrants labeled with Roman numerals. Each quadrant is a mathematically possible $p_\alpha - w_\alpha$ solution/evaluation. Given a (p_α, w_α) pair, we either set α_α to zero to enforce a zero gap, positive pressure contact solution (the red portion of the

vertical axis in Figure 22), or a separation gap, zero pressure non-contact solution (the green portion of the horizontal axis in Figure 22).

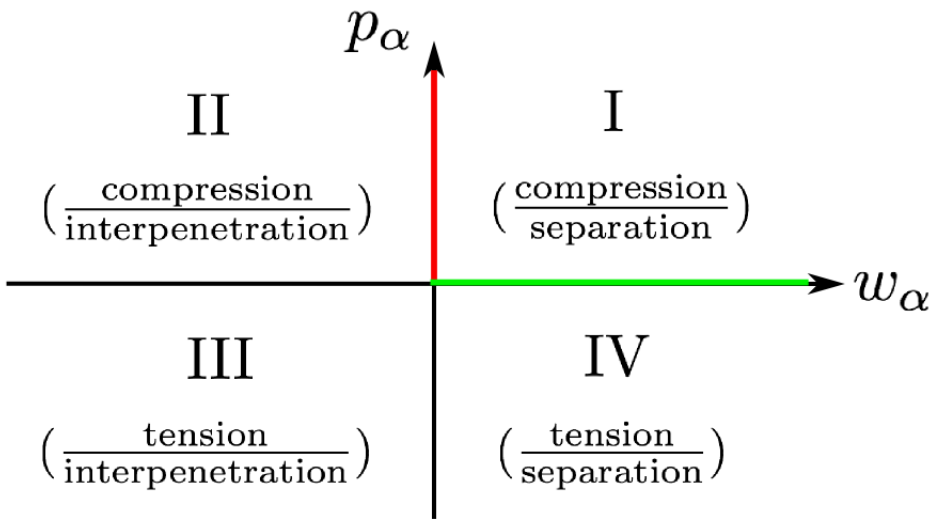

Figure 22: The discrete (p_α, w_α) may fall in quadrants I-IV given a setting of the $(\alpha_\alpha, \beta_\alpha)$ constraint parameters. Given a (p_α, w_α) evaluation, each quadrant dictates a subcycle update for the $(\alpha_\alpha, \beta_\alpha)$ pair. The red and green portions of the vertical and horizontal axes, respectively, represent the two converged contraint enforcements.

Specifically, the update algorithm loops over contact nodes, which store the discrete constant pressure solution for each overlap, and approaches the subcycle parameter update by first considering the current setting of each α_α. If each α_α was set during a mode $= 2$ solution pass using an intermediate value between 0 and 1, then we simply look at the discrete pressure solution to determine the update. If the pressure is positive, then we set α_α to 0, and if the pressure is negative, we set the α_α to 1. We do not consider the w_α evaluation at this point since the intermediate setting of each α_α at a mode $= 2$ solution pass does not exactly enforce the kinematic constraint. Rather, we simply look at the pressure to kick off the update and subsequent subcycling. For all mode $= 4/5$ subcycling solution passes, the alpha-update is simply an update of α_α parameters to either 0 or 1. To do this we first look at the current setting of the α_α parameter (i.e. the setting used to perform the previous mode $= 5$ solve).

If $\alpha_\alpha = 1$, then we have just enforced the constraint based on zero pressure, separation assumption. As a result, we must check the gap to see if this assumption was correct. To do so, we evaluate the gap given the latest mode = 5 solve's incremental nodal displacement solution. If w_α is less than zero, which means that we actually have interpenetration and lie in quadrant III of Figure 22, then we must flip α_α to zero in order to enforce a zero gap constraint due to the interpenetration present in a quadrant III solution. If, on the other hand, we have a positive w_α evaluation then we are in quadrant IV of Figure 22 and have a solution consistent with an adequate enforcement of a non-contact, zero pressure constraint. Note that if $\alpha_\alpha = 1$, then we necessarily obtain a $p_\alpha = 0$ solution. Therefore, for a positive w_α solution, we lie on the green portion of the horizontal axis of Figure 22. Alternatively, if $\alpha_\alpha = 0$, then we have enforced a zero gap constraint, and desire a positive pressure solution. As a result, we proceed by checking the discrete p_α solution. If p_α is negative, then we flip α_α to 1, since a negative pressure solution, regardless of the discrete w_α evaluation, is non-physical. If, on the other hand, p_α is positive, then we simply keep α_α set to 0. It turns out that we do not have to consider the evaluation of w_α here since the gap is implied in the pressure solution. That is, if we have a negative pressure solution, this will reflect a $w_\alpha > 0$ evaluation. As a result, we never land in quadrant I in practice. Similarly, a positive pressure solution with $\alpha_\alpha = 0$ necessarily puts us in quadrant II.

As a parenthetical remark, significant numerical experimentation was conducted using a different α-update strategy. Initially, the thought was that one would *converge* to a particular $(\alpha_\alpha, \beta_\alpha)$ pair and that depending on the discrete (p_α, w_α) solution, one *increments* α_α toward or away from 0 or 1. This method generally worked, and various incrementing strategies were experimented with; however, significant time was spent subcycling. The idea that at any instant, a (p_α, w_α) solution could somehow inform or indicate in an absolute sense which constraint ($\alpha_\alpha = 0$ or $\alpha_\alpha = 1$) one ought to be enforcing for a particular overlap is a somewhat flawed concept since the overlaps are not truly decoupled. As a result, nu-

merical experimentation showed that the direction of the α_α increment changed often during subycling. This poses a possible risk of oscillitory behavior in the α-update where a discrete parameter never actually reaches 0 or 1. The incremental approach appeared to introduce too much flexibility in the solution based on this collection of possibly intermediate contact "stiffnesses." It became obvious that each overlap must enforce one of the two constraints exactly in order to implicitly and properly observe the effect caused by the actual coupling of the contact overlaps through the finite element nodes. Fortunately, the α_α-flipping approach where values take either 0 or 1 proved to be very effective. An initial intermediate setting of all parameters seems to provide an initial stiffness that adequately informs the constraint enforcement at each overlap. Overlaps may experience flipping of the α_α parameter; that is, the first choice for constraint enforcement is not always the right choice. Yet, as Section 11.11 will demonstrate, the flipping back-and-forth at a particular overlap is minimal.

10.4.3 Subcycle Convergence Criteria

Subcycle convergence is determined by whether every (p_α, w_α) pair lies on the red or green portions of the vertical or horizontal axes, respectively, in Figure 22. That is, every contact element exhibits either zero gap with a positive contact pressure or zero pressure with a gap of separation.

For contact problems where the entire contact interface remains in contact through a time step, we will only subcycle once. That is, for problems where every facet-pair exhibits interpenetration where a zero gap constraint is enforced, an initial mode = 2 with intermediate α_α values will be followed by an unconverged mode = 4 solution pass, where all α_α parameters ought to be set to 0, followed by a mode = 5 solve in which case zero gap constraints are enforced throughout. In more complicated contact problems, multiple subcycles will be needed in order to "settle" upon a converged setting of facet-pair-constraints across the

entire contact interface. Once a mode = 4 solution pass has successfully converged, however, the displacement solution is finally used to increment the nodal displacements, the pressure solutions are stored on each contact node, and a mode = 1 solution pass follows in which a residual evaluation is used to determine whether Newton's method has converged for the timestep.

An important and rather subtle note is that the numerical experimentation in this work showed that even for contact interactions where one expects a continuously contacting surface in the presence of globally non-planar geometry, not all facet-pairs exhibited contact constraint enforcement. That is, the interface geometry paired with the surface mesh discretizations and the exact nature of the facet-pair overlaps may be such that a solution where every facet-pair is in contact may not be geometrically possible. The subtlety in this instance is not that a particular facet-pair ought not to be included in a contact interaction from a mechanics standpoint, but ought not to be included from a geometric standpoint in order to arrive at a converged solution. The conclusion is that the subcycling procedure appears to robustly handle geometric characteristics of the discretized contact interface that may otherwise prevent, or at least provide great difficulty in, attaining a converged solution.

10.4.4 Newton Convergence Criteria and the Active Set

The Newton convergence is determined by a residual only evaluation, where the equilibrium residual is checked against some user prescribed tolerance. This is typical in a finite element code for solid mechanics. There are two additional convergence criteria at the mode = 1 level for this contact implementation. One is that the corrections to the incremental nodal displacements for the nodes belonging to facets that form contact pairs must be below a certain tolerance. Additionally, the corrections to the incremental nodal displacements for all other finite element nodes must be below another tolerance.

The first additional criterion is in place to make sure that the contact interface has converged upon a collection of ω_α overlaps that does not change beyond some tolerance between successive Newton iterations. This means that the gap evaluations in a mode = 4 convergence check are computed over ω_α overlaps that are the same as those computed in the previous mode = 2 solution pass, up to some tolerance. The idea is to perform a mode = 1 residual evaluation whose contact integrals, i.e. domains of integration, are commensurate with the pressures from the previous converged subcycle solution. There are some significant practicalities associated with the enforcement of this tolerance, and substantial ways to improve the algorithmic design to achieve the aforementioned goal. These will be discussed later in this section.

The second additional convergence requirement is in place to ensure that no additional facet-pairs come into contact at an otherwise converged Newton iterate. That is, we may have satisfied the convergence requirement using the first geometric tolerance, but the incremental nodal displacements at some portion of the contact interaction outside the current active set may possibly come into contact. To be more explicit, a contact interaction contains two buckets of facets. The active set forms facet-pairs that are said to be in, or possibly may come into, contact. Not all facets in the defined contact interaction must be in the active set. As a result, this geometric tolerance addresses those facets that are not, in order to ensure that they did not come into contact, and therefore possibly missed at an otherwise converged Newton iteration. If this tolerance were not enforced, then the converged timestep could possibly exhibit interpenetration at facet-pairs somewhere outside the computed contact interface. In order to preclude this from happening, a *second* geometric convergence criteria exists at all finite element nodes belonging to facets in all contact interactions that are not part of that interaction's active set. This tolerance is set to the tolerance used on the computed gap to determine facet-pair inclusion into the active set. That is, we have stated that we may include facet-pairs into the active set that exhibit some amount of separation up

to some *numerical* tolerance (typically 1E-1 to 1E-3). We use this same separation tolerance as the second geometric convergence tolerance. The idea is that if facet-pairs were separated slightly beyond this tolerance at the geometry search level, thus not included in the active set, then a converged timestep would never see incremental nodal displacements greater than the separation tolerance at any node belonging to a facet outside the active set. As a result, a "missed" contact pair is not possible.

In practice, there are some issues associated with the geometric tolerances discussed above. While theoretically these tolerances must be enforced in this contact formulation, it turns out that if the tolerances are too strict, then Newton convergence is not obtained. Whereas if the geometric contact tolerances are loosened, then not only is convergence obtained, but smooth contact solutions are still achieved. This highlights the need for further study into appropriate tolerances and algorithmic enforcement in order to achieve the type of convergence behavior that both tolerances aim at enforcing based on the previous two paragraphs' discussion.

The first and more obvious note is regarding the second tolerance. The problems presented in Section 11 do not have the potential for non-active-set pairs to come into contact. This is because the contact interfaces, though exhibiting algorithmically complex or difficult behavior, are not themselves so complex, geometrically speaking. More significant is the fact that the problems presented are such that all overlapping facet pairs are included in the active set, precluding the possibility of further, unforeseen interaction. As the contacting surfaces become more complicated, or should there be multiple regions on one surface that are in the active set, but other regions that are not in the active set, then this tolerance becomes more important. As a result, the contact solutions presented herein do not depend on this tolerance. A proposed update to the numerical implementation centered around this tolerance is to apply the geometric tolerance to nodes on facets that are "near" opposing facets on the other surface. One may imagine a small block indenting a large block. Both opposing

surfaces may define a contact interaction, but only the bottom surface of the top indentor will contact the facets immediately below it on the opposing surface. Obviously there is no need to enforce this geometric criterion at facets that are nowhere near the indentor block. As a result, numerical experimentation did find that arbitrary enforcement of this geometric criteria is in fact too stringent.

The first tolerance described above is applied to the incremental nodal displacement solution at all finite element nodes associated with facets that are used to form contact elements. That is, each component of the nodal displacement vector is checked against this tolerance. When this method was designed, the idea was that if the incremental nodal displacement solution at a node exceeded the tolerance, then we simply take another Newton step, in which case the next iterate's solution will be smaller than the previous. That is, we will be closer to the "exact solution" in a sense, and the correction vector will have very small entries, ideally below the tolerance. The reason for doing so is to obtain a current configuration active set of overlaps that is commensurate with the active set used to obtain the pressure solution. That is, prior to an initial mode = 2 solve, the active set is formed and contact overlaps, specifically ω_αs, are formed. A converged subcycle solution, however, exists with an updated current configuration of the contacting surfaces. If the update using the corrections to the incremental nodal displacements changes the topology of the contact interface, i.e. facet-pairs and their associated overlaps, then we have a new current configuration contact interface that is not commensurate with our pressure solution. Hence, the idea of a converged active set was born. The difficulty, however, is that the idea that we can simply take one more Newton iteration to obtain a solution with very small increments holds for Newton's method applied conventionally to a vector valued function, but appeared to present some difficulties in the context of a contact algorithm. There is the possibility of sensitivity in the contact solution to incremental nodal displacements. As a result, numerical experimentation made it clear that insisting that each node belonging to facets in contact pairs has a displacement

solution below some tolerance may not only be too strict, but may in reality not be possible to achieve given a fixed tolerance. Even still, it is not clear what an appropriate tolerance should be. To fully address this problem, let us first discuss what is being done in the code and the observations seen therein, and propose what should be done as a more rigorous treatment of the active set.

The numerical implementation currently forms the active set prior to a mode = 2 solve, which is followed by mode = 4/5 subcycling. A converged subcycle procedure experiences pressure-gap pairs that adhere to the plot shown in Figure 22. Having said that, the pressure solution is paired with corrections to the incremental nodal displacements that, in general, will modify the configuration of the contact interface. That is, ω_α overlaps will be modified. When we check for Newton convergence in a mode = 1 residual evaluation, i.e. equilibrium convergence, the contact integrals are evaluated using the pressure solution with the original ω_α overlap pairs. As a result, the contact interface discretization being used does not reflect the current configuration as determined from the corrections to the incremental nodal displacements solved for in the prior converged mode = 5 subcycle solution pass. Making the assumption, for the moment, that any change to the median planes is negligible, then the areas of overlap may change, and thus the regions of integration in reality change. If, however, we were to update the facet overlaps for all facet pairs in the current active set, then the pressures would not be commensurate with the new domains of integration. As a result, we are stuck in the middle between pressures with associated overlaps that are not commensurate with the current configuration, and overlaps with pressures that are not commensurate with the current configuration. One idea is to update the facet-pair overlaps, i.e. regions of integration, and recompute the integrals (with the most up-to-date configuration) using a scaled pressure. That is, we scale the pressure with an area ratio between the new and old areas of overlap. While this seems unnecessary, since the integral of the pressure over the original area and the integral of the scaled pressure over the new area ought to

be the same, we must remember that the change in area is likely do to some sliding at the contact interface. As a result, there may be modifications, however slight, to the distribution of contact nodal forces at a contact facet. To address all of the aforementioned points, let us propose an algorithmic change to the convergence criteria. First, let us compute a new active set prior to a mode = 1 residual evaluation rather than prior to a mode = 2 solve. Let us assume for the moment that *all* facet-pairs in the previous active set are in the new active set, no more and no less. If this were the case, let us consider an unmodified active set from the standpoint of facet-pairs as a converged active set rather than insisting on geometric convergence of the facet-pair configurations themselves. Then, to handle any discrepancies in configuration, we perform a pressure scaling based on an area ratio between prior overlap area and current overlap area at all facet-pairs. The overall resultant force will not change, but this allows us to capture any modification to the distribution of nodal forces at each overlap.

The assumption that the collection of facet-pairs in successive active sets is the same may not always be true. For instance, an updated configuration based on a mode = 5 solve may result in the loss of some interacting facet-pairs (for which we may have a pressure solution) and/or the addition of other facet-pairs (for which we do not have a pressure solution). The concept of a geometric tolerance attempts to address this implicitly, but the proposed method in the previous paragraph can more readily handle this reality up front. If we compute a new, or updated active set prior to a mode = 1 evaluation, then we are poised to determine if there are any subtractions or additions to the prior active set, for which we have a pressure and displacement solution. In the case that we acquire new facet-pairs, we do not necessarily want to assume an unconverged active set. Experience studying the subcycling procedure's behavior has shown that very small areas of overlap are often excluded from the "active" contacting facet-pairs in favor of larger areas of overlap. One possible way to handle this is to only include *new* facet-pairs with area ratios between the facet-pair overlap and the

smallest facet's area used to construct the overlap that exceed some tolerance. If the new facet-pair's area ratio exceeds the tolerance, then we include it in the new active set, but conclude that the active set has not yet converged and we move on to another Newton iteration. If the new facet-pair's area ratio falls below the tolerance, we may ignore it (as it likely would not participate in an interaction anyway) and conclude the active set to have converged. Alternatively, if we lose a facet-pair that was in the previous, un-updated, active set and for which we have a pressure solution, then our active set has not yet converged and we are obliged to proceed with another Newton iteration. In reality, forming an over-active active set in combination with smaller increments of deformation suitable to resolve a contact interaction ought to not experience the loss of a facet-pair after a mode = 5 solve. In fact, we specifically want to preclude the possibility that a facet-pair enters and exits the active set on subsequent and continuous Newton iterations and do so by including facet-pairs that exhibit separation up to some tolerance. In general, however, the idea behind the aforementioned logic is to adequately answer the question, did the active set change so much as to invalidate the prior pressure solution? The proposed update to how the numerical implementation handles convergence of the active set addresses this in a more direct and robust manner that explicitly considers and treats the updated configuration of the contact interface with very little additional cost.

As a final note, numerical experiments were run with the first geometric tolerance between 1.E-4 and 1.E-2. Again, it is hard to say how appropriate these values are in relation to modifications of areas of overlap, but these tolerances were less than 1 percent of the smallest facet dimension. Qualitatively, a 1 percent modification to the maximum in-plane dimension of a quadrilateral area of overlap is likely negligible. Nonetheless, some problems were sensitive to the exact value of the tolerance in achieving convergence. What is important to note is that loosening the tolerance *still* led to smooth contact solutions. In fact, any lack of smoothness was more a product of mesh refinement than anything else. As a result,

we may conclude that the application of this tolerance does not adequately achieve what is set out to accomplish in terms of active set "convergence," but the equilibrium residual does not appear to be too sensitive to current configurations of the contact interface that differ, to some extent, from the configuration used to obtain the pressure solution. The problems presented in Section 11 are valid based on the current tolerance implementation since inspection of the numerical results for the problems presented showed that a converged solution did not experience a displacement solution at the subcycle convergence level that modified facet-pair topology to any calculable extent. Additionally, active sets were typically formed including all possible facet-pair combinations in order to test the subcycle procedure, thus precluding the possibility of new facet-pairs coming into contact that were not in the active set to begin with. Furthermore, equilibrium convergence was still obtained, even when using a "loosened" geometric tolerance on finite element nodes associated with contact facet-pairs. Were the geometric discrepancies to be too large between a pressure solution and an updated displacement solution, one would imagine this would adversely affect the global convergence behavior. Yet, more complex interactions or different active sets may make the effect of a converged active set more pronounced. As a result, the aforementioned proposed method seeks to create a more robust solution that does not depend on problem characteristics or user input.

10.4.5 Subcycle Algorithm

The following presents an algorithm in pseudo-code for the contact solution procedure in `Imitor`, including the subcycling procedure. "A.S." stands for the *active set*. Additionally, "Top of Pass" is to denote conditional branching that occurs based on the current MODE at the top of a solution pass *prior* to the element assembly loop. After the element assembly loop, we are at the "Bottom of Pass", where conditional branching controls what is done based on the current MODE. More simplistically, the top of the pass prepares for the element loop. The element loop assembles necessary data, and the bottom of pass does something with that data. This explanation is meant to clarify the general layout of the algorithm below.

Algorithm 2 : Subcycle Algorithm in Pseudo-Code

for $i = 1$ to number of timesteps **do** ▷ time step loop

(inside STEP_SOLN)

Start of Step:
perform velocity extrapolation to obtain initial guess for displacements
compute active contact set (A.S.)
set solution MODE = 1a

for $j = 1$ to max. number of solution passes **do** ▷ solution pass loop

Top of Pass:

if MODE = 1a and j $\neq$ 1 **then** ▷ equilibrium residual evaluation for A.S. = $\varnothing$
if A.S. is null **then**
zero nodal values
else
set MODE = 2
cycle pass loop ▷ convergence not possible for A.S. $\neq \varnothing$
end if
end if

if MODE = 1 and j $\neq$ 1 **then** ▷ equilibrium AND geometric residual evaluation
zero nodal values
end if

if MODE = 2 **then** ▷ Newton iteration w/ solve
compute A.S.
initialize the linear system of equations
allocate contact specific arrays for the "solve"
end if

if MODE = 4 **then** ▷ subcycle convergence evaluation
initialize global length residual vector
end if

if MODE = 5 **then** ▷ subcycle solve
initialize the linear system of equations
end if

for $k = 1$ to number of elements **do** ▷ element loop
▷ (incl. contact elems)

if MODE = 1a **then** ▷ true if A.S. $= \varnothing$ and $j = 1$
compute equilibrium residual contributions for ALL elements
end if

if MODE = 1 **then**
compute equilibrium residual contributions for ALL elements
compute geometry residual at ALL finite element nodes
Note: (for nodes in the A.S. compare corrections to
inc. nodal dofs to small tolerance)
Note: (for nodes not in the A.S. compare corrections
to inc. nodal dofs to larger tolerance, indexed to
A.S. inclusion tolerance)
end if

if MODE = 2 **then**
compute equil. residual contributions for ALL elements
compute equil. tan. stiffness contributions for ALL elements
compute gap evaluations for contributions to the residual
compute gap tan. stiffness contributions for contact elements
end if

if MODE = 4 **then**
contact element gaps (i.e. "w") computed after previous solve
exit element loop
end if

if MODE = 5 **then**
if k is $\leq$ number of continuum finite elements **then**
cycle
else
compute contact elems' residual/tan. stiffness contributions
end if
end if

end for ▷ close element loop

```
        Bottom of Pass:
        if MODE = 1a then
            evaluate equilibrium convergence at all FE nodes
            if converged then
                update global solution and exit solution pass loop
            else
                initialize linear equation object and set MODE = 2
            end if
        end if
        if MODE = 1 then
            evaluate equilibrium AND geometric convergence at all FE nodes
            if converged then
                update nodal dofs and exit solution pass loop
            else
                initialize linear equation object and set MODE = 2
            end if
        end if
        if MODE = 2 then
            store K̄ and F̄ for subsequent subcycle iterations
            solve for incremental nodal disp. corrections and contact pressures
            compute/store the temporary incremental nodal disps. at FE nodes
            set MODE = 4
        end if
        if MODE = 4 then
            evaluate convergence of p-w solution (see Figure 22)
            if converged then
                update the incremental nodal displacement solution
                set MODE = 1
            else
                update (alpha,beta) pair to trend toward a solution on the L
                set MODE = 5
            end if
        end if
        if MODE = 5 then
            solve for incremental nodal disp. corrections and contact pressures
            compute/store the temporary incremental nodal disps. at FE nodes
            set MODE = 4
        end if
    end for                                          ▷ close solution pass loop
end for                                              ▷ close timestep loop
```

10.5 Solution Procedure

This section specifically addresses the mode = 2 and mode = 5 solves. As we have stated, the mode = 2 system of equations is the system of equations formed at a new Newton iteration using an fixed, intermediate setting of all $(\alpha_\alpha, \beta_\alpha)$ pairs. The system of equations is formed using contact elements with intermediate surfaces, ω_α, that are fixed throughout the Newton iterate and its subcycling. This work uses a dense LU factorization to solve the mode = 2 system of equations. Since the global stiffness matrix is sparse, however, a sparse frontal solver may be used for a more efficient and optimal solution procedure. This was not the focus of this work and implementation.

Each mode = 5 subcycle solve is a modification of the original mode = 2 system of equations, where the modification comes from an update to certain, but not necessarily all, $(\alpha_\alpha, \beta_\alpha)$ pairs. A more stable implementation of the subcycling method is achieved using an intermediate setting of all α_α parameters to 0.5 at a mode = 2 solve. Consequently, a mode = 2 solve will not realize exact and appropriate constraint enforcement. As a result, a mode = 2 solve will be followed by one mode = 5 solve where *all* $(\alpha_\alpha, \beta_\alpha)$ pairs are updated. Depending on the contact problem and interface geometry, further subcycling may be needed where each subsequent mode = 5 solve may only see an update of *some* of the $(\alpha_\alpha, \beta_\alpha)$ pairs. The key point is that the system of equations between a mode = 2 Newton system and an updated mode = 5 subcycle system only experiences a nominal modification. Specifically, recall the global system of equations shown in Equation 9.105. The mode = 5 modification to the stiffness matrix will only occur in the bottom $\mathbf{B}\hat{\mathbf{M}}^{\mathrm{T}}$ and $\mathbf{A}$ blocks. Each instance of a $(\alpha_\alpha, \beta_\alpha)$ modification will only affect one row of equations associated with one pressure unknown. As a result, we may write any mode = 5 system of equations as

$$\left[\begin{array}{c|c} \bar{\mathbf{K}} & \hat{\mathbf{M}} \\ \hline (\mathbf{B}+\Delta\mathbf{B})\hat{\mathbf{M}}^{\mathrm{T}} & \mathbf{A}\ +\Delta\mathbf{A} \end{array}\right] \left\{\begin{array}{c} \delta\hat{\mathbf{u}} \\ \hline \hat{\mathbf{p}} \end{array}\right\} = \left\{\begin{array}{c} \mathbf{R} \\ \hline \mathbf{c} \end{array}\right\}. \tag{10.2}$$

For most contact problems, $\bar{\mathbf{K}}$ dominates the size of the stiffness matrix shown in block form in Equation 9.105. As was stated above, not all $(\alpha_\alpha, \beta_\alpha)$ pairs are necessarily modified outside the first mode = 2 to mode = 5 update. Recalling the block structure of Equation 9.105 demonstrates that a mode = 5 update to the system of equations during the subcycling procedure is nominal, where the majority of the blocks in the system do not change at all. Specifically, $\bar{\mathbf{K}}$ and $\hat{\mathbf{M}}$ do not change during the subcycling. Further, the bottom left block will only experience a change in the $\mathbf{B}$ matrix premultiplying the unmodified (and sparse) $\hat{\mathbf{M}}^{\mathrm{T}}$ matrix. Furthermore, recall that $\mathbf{A}$ is a diagonal matrix with the α_α parameters as the diagonal entries. Looking at the structure of the global stiffness matrix and the associated mode = 5 matrix update, the fact that the majority of the stiffness matrix is left unmodified might be exploited in a solution procedure, rather than reforming and refactoring the system every subcycle. To do so, let us write the general global system of equations at a mode = 2 solve as

$$\mathbf{K}\mathbf{u} = \mathbf{f}. \tag{10.3}$$

Furthermore, let us write a mode = 5 update to the system in Equation 10.3 as

$$(\mathbf{K} + \Delta\mathbf{K})\mathbf{u} = (\mathbf{f} + \Delta\mathbf{f}), \tag{10.4}$$

where $\Delta\mathbf{K}$ is shown explicitly in Equation 10.2 and $\Delta\mathbf{f}$ represents the updated residual evaluation at the current Newton iterate with the *updated* $(\alpha_\alpha, \beta_\alpha)$ pairs. To exploit the structure of the updated stiffness matrix $(\mathbf{K} + \Delta\mathbf{K})$ we make the observation that $\Delta\mathbf{K}$ is *low rank*. That is, there are very few nonzero rows in the "modification" matrix. As a result, we seek a method that uses the original factorization of $\mathbf{K}$ to form a factorization of the updated stiffness matrix $\mathbf{K} + \Delta\mathbf{K}$. The following section presents the method used in this implementation.

10.5.1 Matrix Updates

This work has implemented a matrix update method developed in [33] and detailed extensively in [34]. In order to discuss the specific update strategy, some preliminary notation and concepts must first be explained. Equation 10.4 represents an "updated" system of equations that we want to solve. To explore this problem, let us write a general updated system as

$$(\mathbf{A}_0 + \Delta\mathbf{A})\mathbf{x} = \mathbf{b}, \tag{10.5}$$

where $\mathbf{A}_0 \in \mathbb{R}^{nxn}$ with rank, n. $\Delta\mathbf{A} \in \mathbb{R}^{nxn}$, with rank, $r < n$, and $\mathbf{x}, \mathbf{b} \in \mathbb{R}^n$. $\Delta\mathbf{A}$ is referred to as the update matrix and $\mathbf{A}_0$ is an initial factored matrix, where $\mathbf{A}_0 = \mathbf{L}_0\mathbf{U}_0$. $\mathbf{A}_0$ and $\Delta\mathbf{A}$ need not be symmetric. Note that the right hand side in Equation 10.5 does not have a modification, which is a slight departure from Equation 10.4, but is written this way for clarity and to adhere to the literature. Furthermore, for simplicity, we do not consider pivoting in the factorization of $\mathbf{A}_0$, but in principle, LU factorization of $\mathbf{A}_0$ with partial pivoting may be used.

Let us consider the following decomposition of $\Delta\mathbf{A}$:

$$\Delta\mathbf{A} = \sum_{j=1}^{r} \alpha_j \mathbf{x}_j \mathbf{y}_j^{\mathrm{T}}, \tag{10.6}$$

where the j^{th} term of the sum shown in Equation 10.6 is

$$\alpha_j \mathbf{x}_j \mathbf{y}_j^{\mathrm{T}} = \Delta\mathbf{A}_j. \tag{10.7}$$

Given Equation 10.7, we may write Equation 10.6 as

$$\Delta\mathbf{A} = \Delta\mathbf{A}_1 + \Delta\mathbf{A}_2 + \ldots + \Delta\mathbf{A}_r. \tag{10.8}$$

A single term in the update Equation 10.8, which may be written in decomposed form per Equation 10.7, is a rank-one matrix. As a result, Equation 10.6 is really a series of rank-one update terms. Consequently, we may apply a rank-one update procedure "r" times to solve Equation 10.5. [33] specifically presents the method used in this work for rank-one updates for symmetric positive definite matrices. The thesis work presented in [34] treats unsymmetric rank-one updates and presents an algorithm for *multiple* rank-one updates used to solve a system of the form shown in Equation 10.5. The solution procedure used for mode = 5 solves uses the multiple rank-one update method for unsymmtric matrices discussed in [34].

10.5.2 Rank-One Updates

The matrix update theory used to solve Equation 10.4 for a mode = 5 solve, using methods discussed in [34], is presented in this section. Given a decomposition of $\Delta\mathbf{A}$ per Equation 10.6, we may write a single rank-one update as

$$\bar{\mathbf{A}} = \mathbf{A} + \alpha\mathbf{x}\mathbf{y}^{\mathrm{T}}, \tag{10.9}$$

where subscripts have been dropped for simplicity and $\mathbf{A}$ admits the factorization, $\mathbf{A} = \mathbf{LU}$. We may write Equation 10.9 in terms of the factorization of $\mathbf{A}$ as

$$\bar{\mathbf{A}} = \mathbf{L}(\mathbf{I} + \alpha\mathbf{p}\mathbf{q}^{\mathrm{T}})\mathbf{U}, \tag{10.10}$$

where $\mathbf{Lp} = \mathbf{x}$ and $\mathbf{U}^{\mathrm{T}}\mathbf{q} = \mathbf{y}$. Given the following LU factorization (without pivoting for simplicity),

$$\mathbf{I} + \alpha\mathbf{p}\mathbf{q}^{\mathrm{T}} = \tilde{\mathbf{L}}\tilde{\mathbf{U}}, \tag{10.11}$$

the resulting modified factors for Equation 10.9 are

$$\bar{\mathbf{A}} = \mathbf{L}\tilde{\mathbf{L}}\tilde{\mathbf{U}}\mathbf{U} = \bar{\mathbf{L}}\bar{\mathbf{U}}, \tag{10.12}$$

where $\bar{\mathbf{L}} = \mathbf{L}\tilde{\mathbf{L}}$ and $\bar{\mathbf{U}} = \tilde{\mathbf{U}}\mathbf{U}$.

Given a decomposition of a rank-one matrix of the form $\alpha\mathbf{x}\mathbf{y}^{\mathrm{T}}$, the crux of the update lies in the factorization of $\mathbf{I}+\alpha\mathbf{p}\mathbf{q}^{\mathrm{T}}$ of Equation 10.11 and efficient matrix multiplication to form $\bar{\mathbf{L}}$ and $\bar{\mathbf{U}}$. The work presented in [34] actually presents efficient algorithms to compute both of these items. The algorithm presented in this work forms $\mathbf{p}$ and $\mathbf{q}$, factors Equation 10.11, and performs the aforementioned matrix multiplications to form the updated factors. The flop count for one rank-one update is approximately $4\mathcal{O}(n^2)$. If the number of nonzero rows in $\Delta\mathbf{A}$, i.e. the number of modified rows in the global stiffness matrix, is r, then the flop count is approximately $4r\mathcal{O}(n^2)$, which is compared to $\frac{2}{3}\mathcal{O}(n^3)$ for a full matrix decomposition at a given mode = 5 solve. If $r << n$, then the rank-one update is a much more efficient solution strategy. Note that for $r \to n$, the number of contact elements would have to exceed the number of finite element nodes in a simulation. That is, each finite element node has three corresponding rows in the global stiffness matrix for 3D problems. Even if there were as many contact elements as finite element nodes, which may be physically impossible for any given simulation, there would still be three times less rows in the global stiffness matrix that may be modified in a new mode = 5 solve. In any case, any given mode = 5 solve will likely not see all possible contact rows undergo a modification. Finally, more demanding subcycling and row modification happens under very difficult quasi-static contact problems, and in general, for a large class of less difficult problems, the subcycling will be minimal. As a result, a rank-one update strategy, like the one presented here, is essential to an efficient solution strategy.

10.5.3 Decomposition of the Update Matrix

A final algorithmic consideration is the decomposition of an update matrix, $\Delta\mathbf{A}_j$. We know that $\Delta\mathbf{A}$, the full modification matrix, will only contain nonzero rows where there have been modifications in $\mathbf{B}$ and $\mathbf{A}$ associated with an update of β_α and α_α parameters, respectively. As a result, we can take each constituent update matrix, $\Delta\mathbf{A}_j$, as the zero matrix save for an individual nonzero row corresponding to a given contact element with modified subcycle parameters. Therefore, we perform a rank-one update on a row-by-row basis for all modified rows. To form the decomposition of $\Delta\mathbf{A}_j$, let j be the row number in the global stiffness matrix corresponding to a modified row. We use the j^{th} modified row and the j^{th} unit vector and form an outer product. Let the j^{th} modified row vector in the global stiffness matrix be denoted $\mathbf{a}_j$ and the j^{th} unit vector denoted $\mathbf{e}_j$. Then,

$$\Delta\mathbf{A}_j = \alpha_j \mathbf{e}_j \mathbf{a}_j^{\mathrm{T}}, \tag{10.13}$$

where $\mathbf{a}_j^{\mathrm{T}}$ denotes the modified row in row-vector form and α_j is taken to be 1. With this development in mind, we may write Equation 10.6 as

$$\Delta\mathbf{A} = \sum_{j=1}^{r} \alpha_j \mathbf{e}_j \mathbf{a}_j^{\mathrm{T}}. \tag{10.14}$$

Other decompositions are possible (discussed in [34]), but this decomposition proved useful, requiring no additional computation to perform, and easily conforming to the row-modification pattern observed in the update matrix.

10.5.4 Notes on Sparse Rank-One Updates

As previously mentioned, extending the solution procedure to a sparse implementation was not the focus of this work. In principle, however, there is no aspect of the aforementioned solution procedures in the dense setting that cannot be extended to the sparse setting. Additionally, the rank-one updates do not need to use an LU factorization of the original matrix. A sparse Cholesky factorization, for example, may be used. The work presented in [34] discusses sparse impelementations of multiple rank-one updates in a detailed fashion. As a result, the implementation of multiple rank-one updates for the subcycle solution procedure presented in this work is a strong proof-of-concept for a more advanced sparse implementation, and demonstrates the efficiency of such methods.

10.6 Discussions on a Friction Implementation

This work does not include a friction implementation; however, some thought has been given as to how to incorporate friction within the subcycle procedure setting. This section is included for the purposes of discussing the key points of this development. The ideas contained herein have not yet been implemented, but the discussion is one of constraint enforcement, to which friction nicely conforms to and the subcycle procedure has been shown to effectively handle.

10.6.1 The Tangential Traction Term

The tangential contact traction term is given in Equation 9.18). In that equation, $\hat{\boldsymbol{\tau}}_\alpha$ is a piecewise constant tangential traction component defined on each discrete contact overlap ω_α. We may define an in-plane, orthonormal basis local to each median plane as $\{\mathbf{e}_1^\alpha, \mathbf{e}_2^\alpha\}$ such that the tangential traction component is defined as

$$\hat{\boldsymbol{\tau}}_\alpha = \hat{\tau}_1^\alpha \mathbf{e}_1^\alpha + \hat{\tau}_2^\alpha \mathbf{e}_2^\alpha, \tag{10.15}$$

where $(\hat{\tau}_1^\alpha, \hat{\tau}_2^\alpha)$ are the two unknown tangential traction constants. The friction term introduces another contact integral term to the contact portion of the finite element residual,

$$\hat{R}_{ja} = -\int_{\gamma_c} \hat{t}_j \varphi_a^{(i)} dA \;\; = -\int_{\gamma_c} \hat{t}_j^{(n)} \varphi_a^{(i)} da \;\; - \int_{\gamma_c} \hat{t}_j^{(t)} \varphi_a^{(i)} da, \tag{10.16}$$

where,

$$-\int_{\gamma_c} \hat{t}_j^{(t)} \varphi_a^{(i)} dA = \;\; -\sum_\alpha \int_{\omega_\alpha} (\hat{\tau}_1^\alpha e_{1j}^\alpha + \hat{\tau}_2^\alpha e_{2j}^\alpha) \varphi_a^{(i)} da. \tag{10.17}$$

In terms of contact unknowns, there is a modification to the contact nodal force mapping, where for each overlap there are up to three linear polynomial terms for the normal traction

component and two terms for the constant tangential traction component. The vector of unknowns at each discrete overlap may be written as, $[q_0^\alpha \;\; r_1^\alpha \;\; r_2^\alpha \;\; \hat{\tau}_1^\alpha \;\;\; \hat{\tau}_2^\alpha \;]^{\mathrm{T}}$. The additional contact integrals from the tangential terms will involve integrals of shape functions over each ω_α overlap. These integrals will have been computed for the normal contact enforcement, so incorporating the tangential unknowns requires a simple modification of $\mathbf{M}$ and the global vector of unknowns. The addition of the tangential unknowns is not expected to introduce instability into the system of equations. As a result, if a full linear pressure polynomial is used for the normal contact enforcement, special care must be taken to address these columns of $\mathbf{M}$ only when forming a full column rank $\hat{\mathbf{M}}$, leaving the columns associated with the tangential traction terms alone. This will require some modifications to the algorithm used to form a full rank $\hat{\mathbf{M}}$, but in principle the modifications are straightforward.

10.6.2 Friction Constraints

Let us define the scalar valued slip defined at each discrete overlap, ω_α, as

$$\hat{s}_\alpha = \frac{1}{|\omega_\alpha|} \int_{\omega_\alpha} \mathbf{m}_\alpha \cdot (\hat{\mathbf{u}}^{(2)} - \hat{\mathbf{u}}^{(1)}) da, \tag{10.18}$$

where $\mathbf{m}_\alpha$ is the slip vector at ω_α, $\mathbf{m}_\alpha \cdot \bar{\mathbf{n}}_\alpha = 0$, and $\hat{\mathbf{u}}^{(i)}, i = 1, 2$ is the incremental nodal displacement vector on contact surface i. Equation 10.18 is the scalar valued slip where the difference in the incremental nodal displacement on each side is projected onto the tangential slip vector. Defining all slip as positive, the slip constraint, or tangential kinematic constraint, is

$$\hat{s}_\alpha \geq 0. \tag{10.19}$$

If $\hat{s}_\alpha = 0$, then we have a stick condition, and if $\hat{s}_\alpha > 0$, then slipping occurs. The intention is to implement a Coulomb friction model. As such, there are conditions on the tangential

traction in the case of stick and slip. These are

$$\hat{s}_\alpha = 0, |\hat{\boldsymbol{\tau}}_\alpha| < \mu_0 p_\alpha, \text{ (stick)}, \tag{10.20}$$

and

$$\hat{s}_\alpha > 0, \hat{\boldsymbol{\tau}}_\alpha = \mu p_\alpha \frac{\hat{\mathbf{s}}}{|\hat{\mathbf{s}}|}, \text{ (slip)}, \tag{10.21}$$

where μ_0 and μ are the coefficients of static and dynamic friction, respectively, p_α is the contact pressure at ω_α, and $\hat{\mathbf{s}} = \hat{s}_\alpha \mathbf{m}_\alpha$.

Clearly we have introduced another inequality constraint in Equation 10.19. Similar to the gap constraint, we introduce a slack variable in order to convert Equation 10.19 to an equality constraint. This is written as

$$s_\alpha - \hat{s}_\alpha = 0, \tag{10.22}$$

where s_α is the scalar valued slack variable. Recall that introducing a slack variable in effect introduces another set of unknowns. For this reason, and similar to how the normal gap constraint was treated, we want to introduce a linear relationship between the tangential slack variable and the magnitude of the tangential traction. This relationship is as follows:

$$\lambda_\alpha(\tau_\alpha - \mu p_\alpha) - \nu_\alpha s_\alpha = 0, \tag{10.23}$$

where for simplicity of notation, $\tau_\alpha = |\hat{\boldsymbol{\tau}}_\alpha|$ is the magnitude of the tangential traction. Similar to the $(\alpha_\alpha, \beta_\alpha)$ parameters, we introduce two new parameters controlling the friction constraint enforcement, $(\lambda_\alpha, \nu_\alpha)$. Similar to α_α, we let $\lambda_\alpha \in [0, 1]$ where if $\lambda_\alpha = 0$ and $\nu_\alpha > 0$, $s_\alpha = 0$, and if $\lambda_\alpha = 1$ and $\nu_\alpha = 0$ then $(\tau_\alpha - \mu p_\alpha) = 0$, which is to say the magnitude of the tangential traction solution at ω_α must equal the magnitude of the dynamic friction traction, which was introduced in Equation 10.21. This simply shows how the $(\lambda_\alpha, \nu_\alpha)$ parameters affect the linear relationship in Equation 10.23. To get to the actual constraint enforcement,

we must multiply Equation 10.22 through by ν_α to obtain the following expression,

$$\nu_\alpha s_\alpha - \nu_\alpha \hat{s}_\alpha = \lambda_\alpha(\tau_\alpha - \mu p_\alpha) - \nu_\alpha \hat{s}_\alpha = 0. \tag{10.24}$$

Equation 10.24 represents the final form of the equality constraint for friction. The constraint enforcement requires subcycling over $(\lambda_\alpha, \nu_\alpha)$ pairs (though the exact expression for ν_α has not been derived, it will depend on λ_α similar to how β_α depended on α_α). Using Equation 10.24, we see that if $\lambda_\alpha = 0$ and $\nu_\alpha = 1$, then we enforce $\hat{s}_\alpha = 0$, which is the stick condition. Here we do not actually enforce any condition on the tangential traction. Rather, similar to how the zero normal gap constraint was enforced, if we get back a tangential traction whose magnitude is greater than μp_α, then we "flip" λ_α to 1. Then, if $\lambda_\alpha = 1$ and $\nu_\alpha = 0$, we enforce the traction condition $(\tau_\alpha - \mu p_\alpha) = 0$, which is the slip condition. If in fact $\tau_\alpha \neq \mu p_\alpha$, then we flip λ_α to the stick condition. While this methodology has not been implemented, one can see that the enforcement of the friction constraints falls into the form of the subcycle procedure very nicely. A final note on the friction constraints is that they are only enforced at contact overlaps where $\alpha_\alpha = 0$. That is, only at overlaps where a zero-gap contact constraint is currently being enforced during the subcycle procedure.

10.6.3 A Note on the Slip Vector

One final note concerns the slip vector, $\mathbf{m}_\alpha$. As we enforce the friction constraints at any given subcycle, we will not a priori know the direction of slip. In fact, the updated incremental nodal displacements will be as they are, and so we must predict $\mathbf{m}_\alpha$. Let us refer to the predicted slip vector as, $\tilde{\mathbf{m}}_\alpha$. The idea is that a converged contact solution involving friction will be such that $\tilde{\mathbf{m}}_\alpha = \mathbf{m}_\alpha$. The exact methodology for predicting the slip vector is still under development. Furthermore, one of the main algorithmic tests will be to understand how the prediction of the slip vector affects the behavior of the contact enforcement

throughout subcycling.

11 Numerical Examples

A suite of numerical examples is presented that tests and demonstrates the efficacy and robustness of the contact algorithm presented in this work. The first examples are contact patch tests with conforming and non-conforming meshes at the contact interface. These tests are designed to demonstrate equal-and-opposite contact nodal forces at the contact interface, as well as a constant stress solution in each body. Then, a series of stacked cantilever beam problems is presented whereby the displacement of the top beam induces the deformation of the bottom cantilever beam through the contact interaction at the common interface. Each beam uses a hyperelastic material model. These problems demonstrate frictionless sliding contact, and showcase the algorithm's ability to handle complex interface geometry and active set. Following this is another series of structural mechanics problems whereby two beams are stacked on top of each other, both with fixed-fixed end conditions. The beams are deformed into double curvature, again demonstrating sliding contact and the algorithm's ability to handle a complex interface defined by localized regions of high curvature. In the aforementioned structural mechanics problems, the contact interfaces are flat in their original configuration. The next suite of problems further exemplifies the contact algorithm's capabilities by introducing a perturbed contact interface in the stacked cantilever problem. Lastly, a variety of large sliding problems demonstrate the algorithm's ability to effectively handle multiple interface regions coming in and out of contact through the course of a deformation process. To conclude this section, two of the numerical examples are revisited and contact subcycle performance is discussed in detail.

11.1 Patch Tests

The contact patch test presented herein is a boundary value problem whose geometry is composed of two stacked cubes. With the vertical z-axis passing upward through the two cubes, a vertical displacement boundary condition is prescribed in the negative z-direction at all of the nodes located on the top surface of the top cube. The remaining boundary conditions are such that a constant stress solution should be achieved in both cubes. What makes this a displacement driven *contact* patch test is that the deformation and resulting stresses in the bottom cube are induced through the contact interaction at the common interface. In the following patch tests, two cubes with a conforming mesh at the interface are presented, as well as two cubes with a non-conforming mesh at the contact interface. A hypoelastic material model is used for both cubes in both cases with E = 1.E6Mpa and $\nu = 0.25$ being the Young's modulus and Poisson's ratio, respectively. The conforming and non-conforming cases show a constant stress solution with equal-and-opposite contact nodal forces at the nodes comprising the contact interface. This condition constitutes passage of the contact patch test.

11.1.1 Conforming Meshes

Two unit meter cubes with equal discretizations are stacked one on top of the other. A displacement boundary condition in the downward direction is specified at the top nodes at the top surface of the top cube. The nodes at the bottom of the lower cube are restrained in the upward direction and allow for transverse displacement. Additionally, there is restraint against rigid body motion. These support conditions are such that a constant stress solution may be obtained. The response of the bottom cube is induced through the contact interaction at the common interface.

Figure 23 shows a displacement contour plot at the final configuration of the two cubes. The top displacement boundary condition was -1E-3. The T_{33} stress solution (in MPa) is shown in the contour plot in Figure 24. This figure exhibits the same constant stress solution in both cubes. As a result, we can conclude that the method passes the contact patch test for a conforming contact interface.

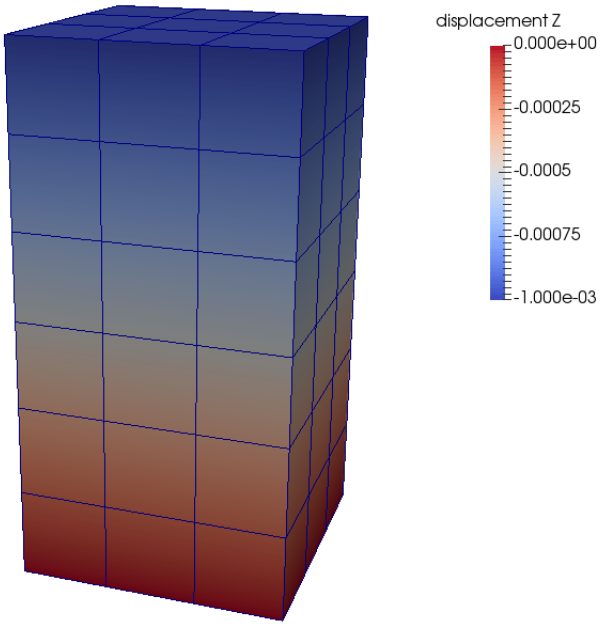

Figure 23: Final configuration z-displacement contour plot of a two-block patch test with a conforming contact interface.

As a note, the work in [9] is also designed to pass the contact patch test. This is an advantageous characteristic of face-on-face methods. Traditional node-to-segment methods, as noted in that work, may pass the contact patch test under certain conditions, or may not pass the contact patch test at all. This method is designed to ensure traction *and* nodal force equilibrium at the contact interface, which are characteristics of passage of the contact patch test.

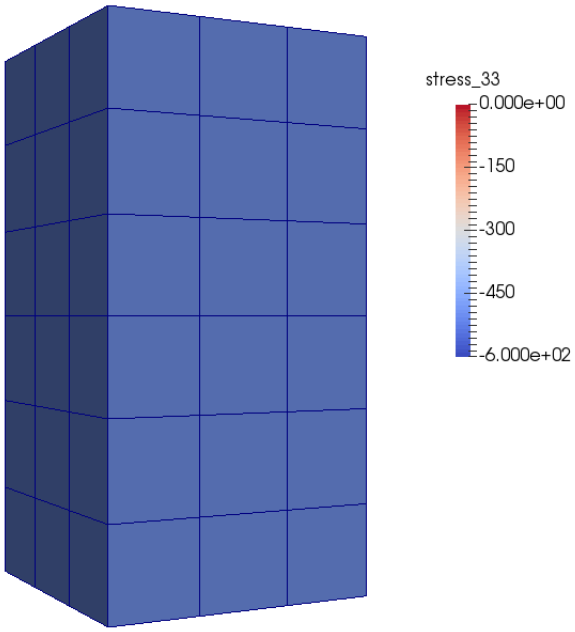

Figure 24: Final configuration T_{33} stress contour plot for the conforming patch test.

11.1.2 Nonconforming Meshes

Figures 25 and 26 show the contact surface meshes for the bottom cube and top cube, respectively. The meshes are clearly non-conforming. Similar to the conforming patch test, the bottom cube has unit dimensions, and the top cube has unit height while the (x,y) dimensions of the top cube are slightly less than unit such that no facet's segment on the top cube's contact surface is coincident with a facet's segment on the bottom cube's contact surface.

Identical to the conforming patch test, a displacement boundary condition was specified at all nodes on the top surface of the top cube. This was set to 1E-3. Figure 27 shows a displacement contour plot in the 3-direction showing the final configuration of the two cubes. Additionally, Figure 28 shows a T_{33} stress contour plot (in MPa). The first observation is that a constant stress solution is obtained in each block. Because the cross-sectional areas of the two blocks at the contact interface are slightly different, there is a small difference in

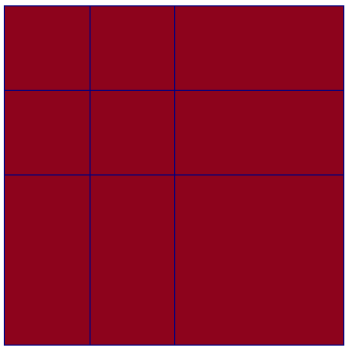

Figure 25: The bottom contact surface mesh for the nonconforming patch test

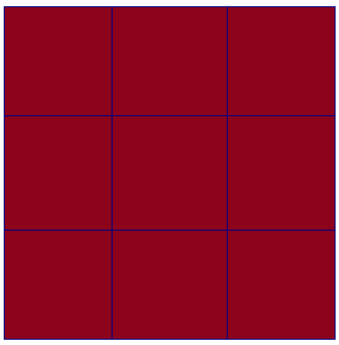

Figure 26: The top contact surface mesh for the nonconforming patch test

the stress solution in each block deriving from the contact interaction at the interface. In all, however, the two cubes pass the contact patch test.

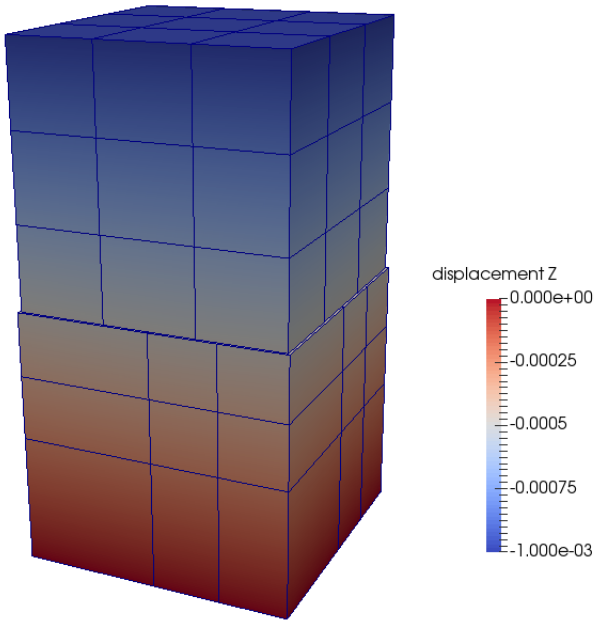

Figure 27: Final configuration z-displacement contour plot of the two block patch test with non-conforming contact surface meshes.

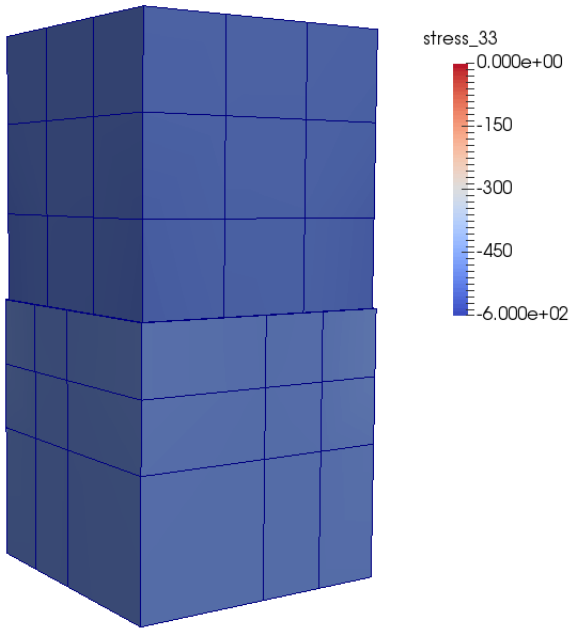

Figure 28: Final configuration T_{33} stress contour plot for the nonconforming patch test.

11.2 Contacting Cantilever Beams

This problem is designed to exhibit sliding across complex interface geometry. Two cantilever beams are "stacked" one on top of the other with a contact interface between the two. The length of each beam if 4.0m and the height of each beam is 0.5m, making the height of the total assemblage 1.0m. The width of each beam is 0.25m. Each beam consists of one eight node hex element through the thickness (i.e. width) and two "hex8" elements through the depth (i.e. height). Problems with conforming meshes at the contact interface have sixteen elements through the length of each beam, and problems with non-conforming meshes across the interface have sixteen and seventeen elements through the lengths of the bottom and top beams, respectively. Figure 29 shows the initial configuration, in perspective, of the two cantilever beams with conforming meshes across the interface, which is located at mid-height of the total assemblage. Figure 30 shows the initial configuration, in perspective, of the two cantilever beams with non-conforming meshes across the contact interface. The nodes at

each beam's left hand support are fully fixed. Two flavors of this problem are presented, one with a specified tip displacement via a displacement boundary condition prescribed at the tip nodes of the top cantilever, and the other with a uniformly distributed traction on the top surface of the top cantilever. In all stacked cantilever problems a hyperelastic material model (see Section 3.5) is used with the following parameters:

$$D_1 = 87083.0,\ C_1 = C_2 = 20096.2.$$

These material parameters are in units of MPa where D_1 is half the bulk modulus and $C_i, i = 1, 2$ is one-quarter the shear modulus of structural steel. These are the material parameters for the Mooney-Rivlin hyperelastic material model for all stacked cantilever problems unless otherwise noted. The aforementioned loading cases are presented in the two subsequent sections. All displacement contour plots are in units of meters and all stress component contour plots in units of MPa.

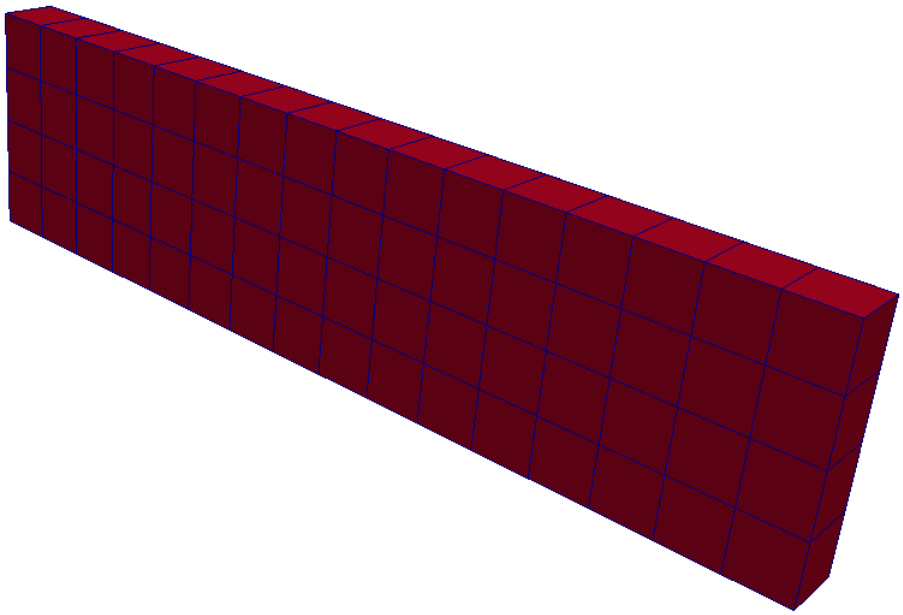

Figure 29: Initial configuration of the conforming stacked cantilever beams.

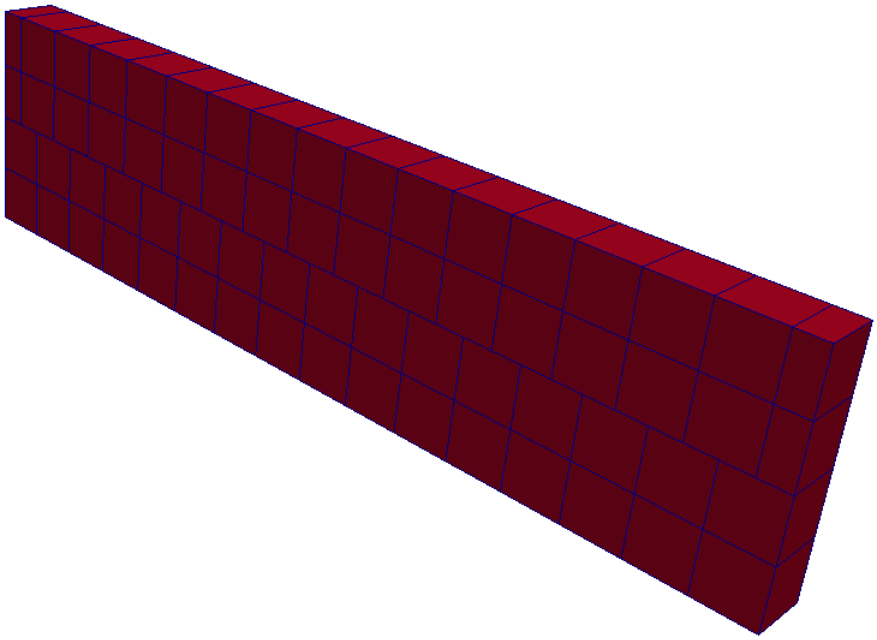

Figure 30: Initial configuration of the nonconforming stacked cantilever beams.

11.2.1 Traction Boundary Condition

This numerical example applies a uniformly distributed traction along the top facets of the top cantilever beam. For all problems of this type, the traction is applied normal to the top surface of the top beam. A diagram showing the initial configuration of the two bodies and associated boundary conditions is shown in Figure 31 where $\bar{\mathbf{t}} \cdot \mathbf{e}_3 = 500$ MPa where $\bar{\mathbf{t}}$ is the prescribed Cauchy traction and $\mathbf{e}_3$ is the unit basis vector in the 3-direction. The loading condition results in finite deformation of the two cantilevers and sliding at the contact interface.

Displacement contour plots showing the final configuration of the cantilevers for both the conforming and non-conforming interface meshes are shown in Figures 32 and 33, respectively. What is interesting about this problem is that very little subcycling occurs during each Newton step. The uniformly distributed traction boundary condition results in nearly all overlapping facet-pairs to remain in compressive contact. Yet, the presence of interface

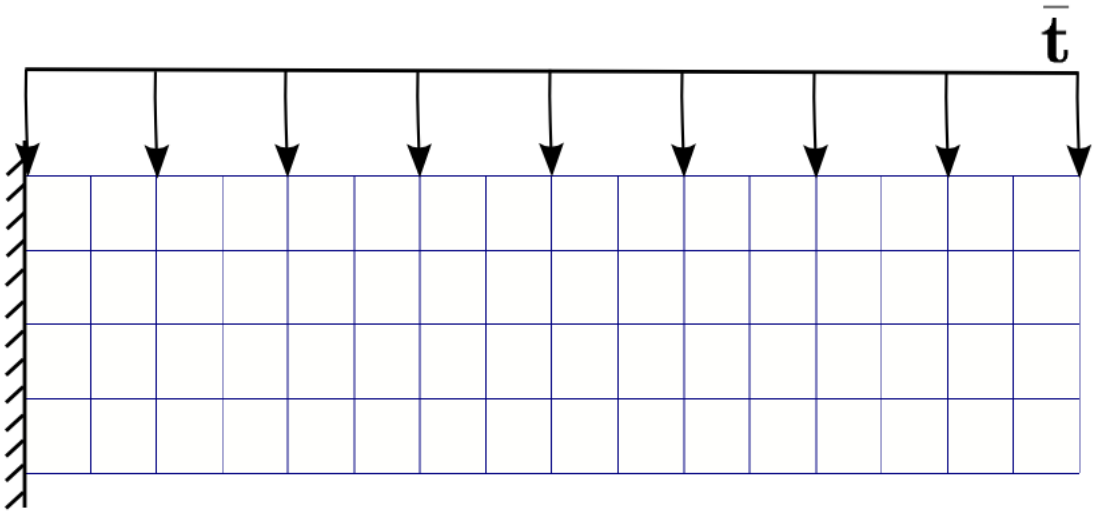

Figure 31: Diagram showing the loading and support condition for the stacked cantilever contact problem.

sliding in this problem highlights a feature of the subcycling method that is worth elaborating on.

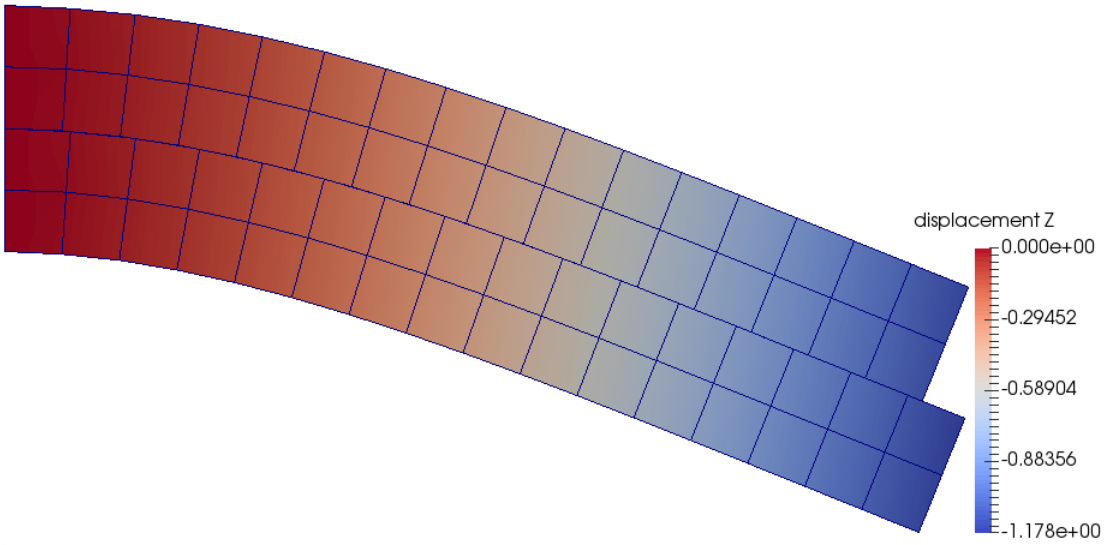

Figure 32: Final configuration z-displacement contour plot of the conforming stacked cantilever problem.

At the very beginning of the simulation there is nearly zero sliding, such that one contact

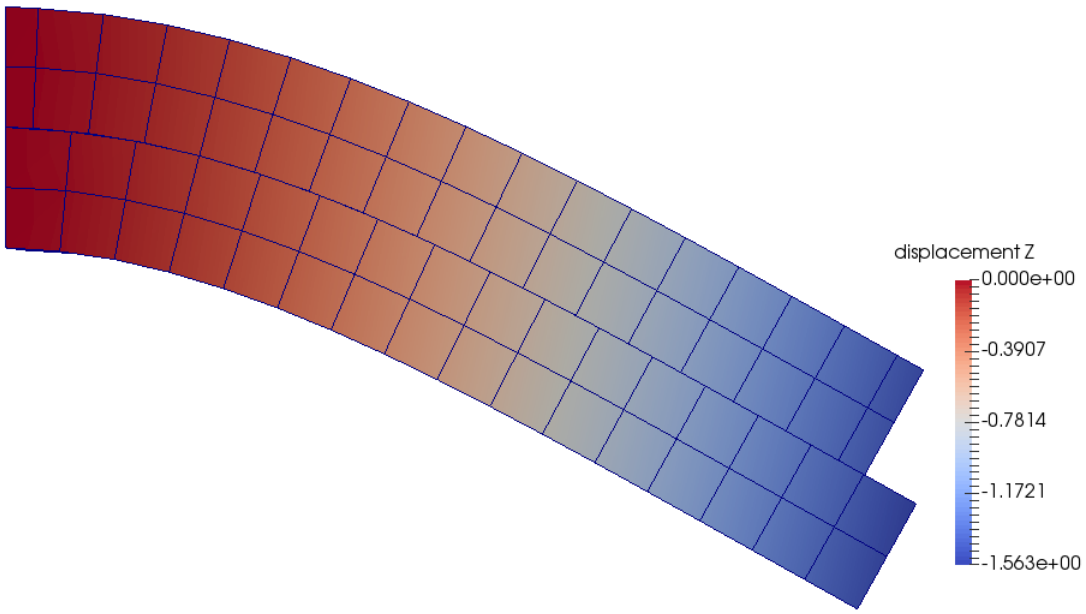

Figure 33: Final configuration z-displacement contour plot of the nonconforming stacked cantilever problem.

facet has not slid to interact with two adjacent contact facets on the opposing surface. As a result, all facet overlaps are typically found to be in contact by the subcycling method. When sliding occurs with the presented conforming and non-conforming contact interface meshes, however, there is always a very small facet overlap adjacent to a sizeably larger facet overlap as one facet slides between two adjacent facets on the opposing contact surface. One observation made throughout the numerical experimentation phase of this work is that it may not be possible to find a converged solution where all facet pairs, both "large" and "small", are in compressive contact. While the physical solution may be continuous commpressive contact across the entire contact interface, the kinematically "possible" solution may and, in fact, often does exclude the small overlaps from zero mean gap constraint enforcement. The subcycling method presented in Section 10.4 handles this very naturally, whereas if the subcycle procedure were not performed and zero mean gap constraints were enforced at all facet overlaps in the active set, which would include the aforementioned "small" facet overlaps, some tensile pressure solutions would result. Again, the subcycling procedure

obviates this dilemma. A further point must be made, which is that the two meshes used in this problem have somewhat similarly sized elements across the contact interface. In the presence of sliding then, there is a larger discrepancy in overlap area between a facet on one surface that just slid into contact with a new facet on the other surface and the area left over from the original, pre-slip, facet-facet pair. Experimentation has shown that the larger area tends to dominate the solution with compressive contact pressures and zero mean gap enforcement occuring at these elements. Furthermore, in the presence of mesh refinement, a kinematically "possible" solution is more likely to involve more facet overlaps with smaller contact facet on each surface of the interface. Lastly, this problem does involve moderate curvature near the support. It is often difficult to resolve contact in the presence of curvature over surfaces with larger contact facets. One aspect of this work that eases this problem is in fact the integral formulation of the zero mean gap constraint, but one aspect of this work that significantly enhances contact enforcement under these conditions is the use of the subcycling method. This becomes more significant and is discussed in more detail in the next cantilever example.

11.2.2 Displacement Boundary Condition

A displacement boundary condition is specified at the nodes at the tip of the top cantilever, which drives the majority of the deformation in this problem. The displacement of the bottom beam is induced by contact across the interface. Additionally, a Cauchy traction boundary condition is applied at the right hand side of the bottom beam along the outermost surface of the bottom-most tip element. A diagram showing the boundary condtions as such is shown in Figure 34, where $\bar{u}_3 = -2.0$m and $\bar{\mathbf{t}} \cdot \mathbf{e}_3 = 100$ MPa where $\bar{\mathbf{t}}$ is a prescribed Cauchy traction and $\mathbf{e}_3$ is the unit basis vector in the 3-direction. The prescribed traction is applied simply to induce more curvature in the beam assemblage, resulting in a more complex interface geometry over which relative sliding between the two beams will occur.

The large deformation of the beams results in complex interface geometry. The reason

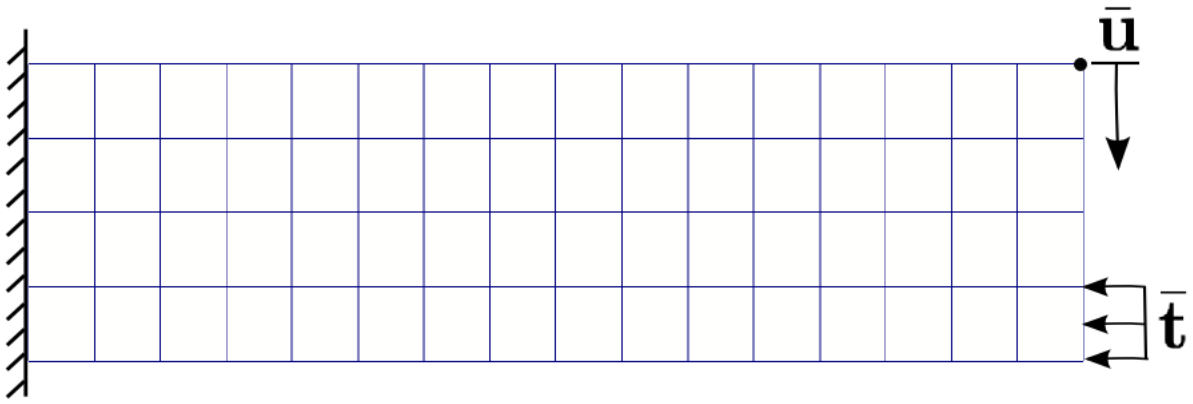

Figure 34: Diagram showing boundary conditions and support conditions for this stacked cantilever contact problem.

the contact interface is "complex" is two-fold. The large deformation experienced in this problem creates a high degree of curvature in both beams, resulting in a curved contact interface, as well as relative sliding between the two beams across the interface. Strict point-wise contact enforcement, such as that found in a node-to-segment formulation, would result in locking. The integral gap formulation used in this method, however, results in contact enforcement even across a "moderately" discretized contact interface experiencing a high degree of curvature *and* in the presence of sliding. Secondly, deformation driven by a displacement boundary condition at the tip of the top beam results in the top beam "kicking" up creating a small amount of localized separation across the contact interface throughout the deformation process.

The initial configuration of the two beams, of which the conforming case is shown in Figure 34, has the two beams resting one on top of the other with zero gap. As a result, the contact algorithm will form contact elements at all facet-facet overlaps. As the deformation process proceeds, certainly some of those contact elements will experience compressive contact, but the "kicking" up of the top cantilever beam will create a localized region of separation. Interestingly enough, that separation is small enough to easily fall within the

"separation tolerance" used to determine which facet-pairs to include in the active set. That is, even facet pairs with a "small" amount of separation will be included in the active set. The idea is that the next increment of deformation may send the previously separated facet pair into contact. If that pair, though exhibiting a small amount of separation, were not included in the active set, then that facet interaction would be missed. The localized region of separation caused by the deformation process is such that the gap of separation is small enough and all possible facet-pairs are always included in the active set. The reason this is significant is that if subcycling over the contact constraints were *not* performed, then the facet-pairs in the active set exhibiting a small amount of separation will be forced together with a resulting tensile contact pressure, which is unphysical. This work found that it is not possible in all scenarios to know, a priori, exactly which facet-pairs will *stay* in contact, come *into* contact, or come *out of* contact within an increment of deformation. To adequately address this, a possibly overpopulated active set of facet-pairs is considered (overpopulated in the sense that facet-pairs with "small" separations are thrown into the active set), and the subcycling method presented in Section 10.4 is performed to determine the appropriate active constraints for the collection of interacting facet-pairs.

Figure 35 and Figure 36 are displacement contour plots showing the final configuration of the conforming and non-conforming interface problems, respectively. Notice the interface slip induced by the deformation where in both cases one facet slides entirely onto an adjacent facet at the contact interface. Examples of a region of localized separation is shown in Figure 37, while Figure 38 shows a localized region where each facet pair is *in* contact. The successful determination of which facet-pairs have zero mean gap contact enforcement is the product of the subcycling procedure presented in Section 10.4.

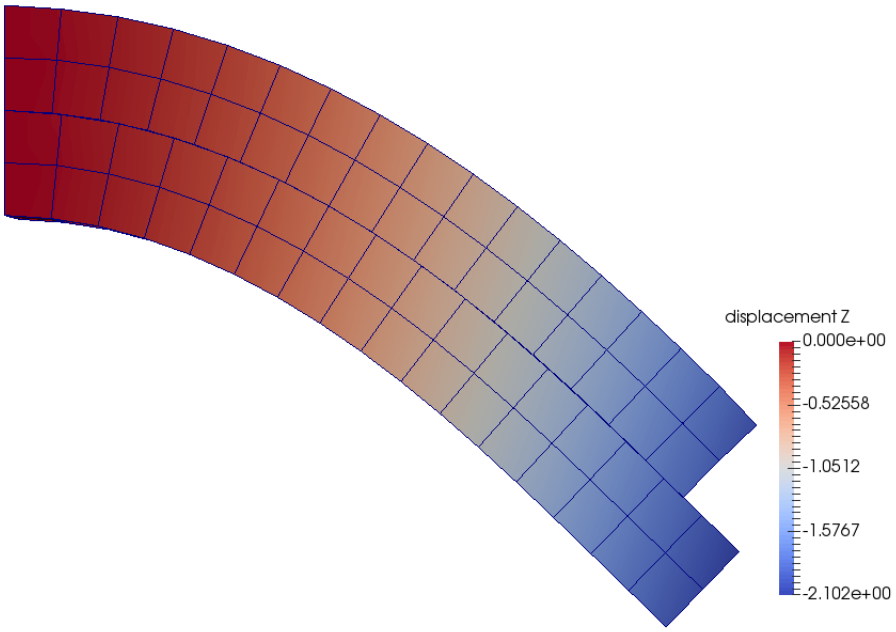

Figure 35: Final configuration z-displacement contour plot of the conforming stacked cantilever problem with displacement boundary conditions.

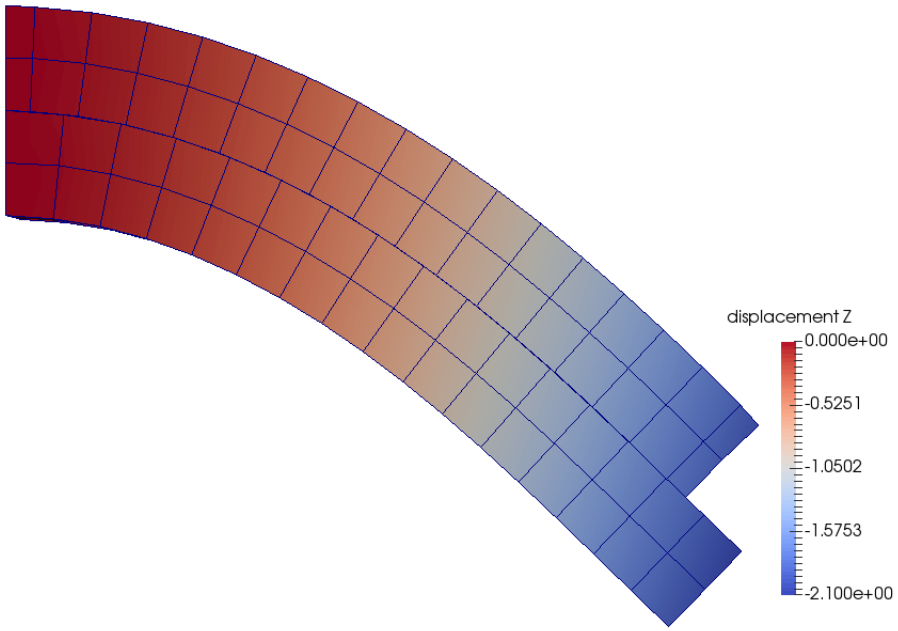

Figure 36: Final configuration z-displacement contour plot of the nonconforming stacked cantilever problem with displacement boundary conditions.

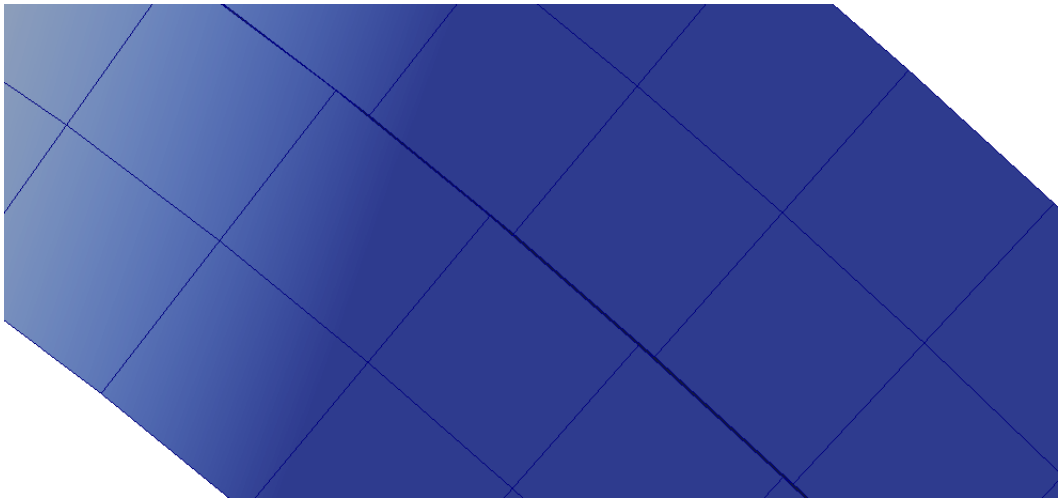

Figure 37: Closeup view of the contact interface in a localized, non-contacting region.

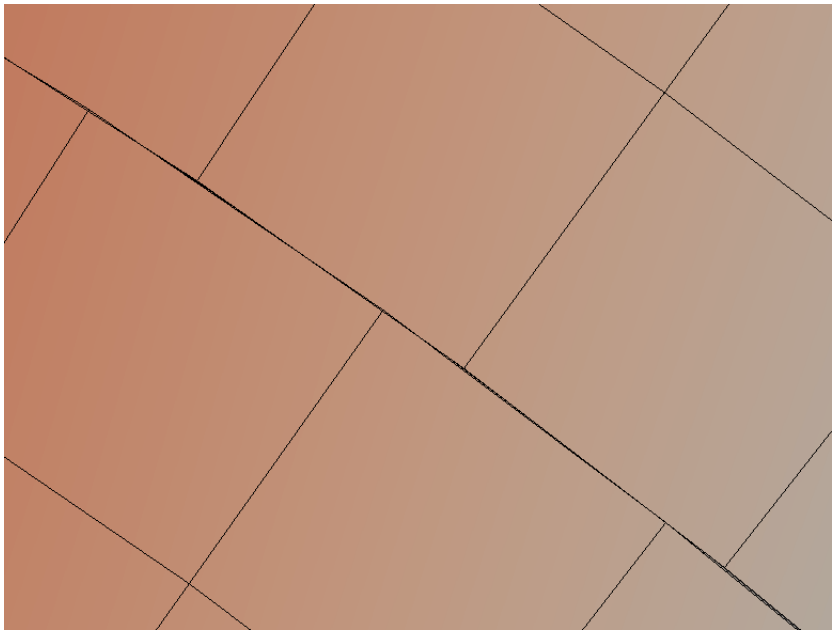

Figure 38: Closeup view of the contact interface showing a localized region of contact. Note that the integral form of the gap constraint allows for some amount of nodal interpenetration, as evidenced by this figure.

11.3 Contacting Fixed-Fixed Beams

This structural mechanics problem is a stacked, fixed-fixed beam problem where one end experiences a downward displacement in the negative z-direction as well as a longitudinal displacement in the negative x-direction, inducing compression in the two beams. Figure 39 shows the boundary conditions and support conditions for this problem. The boundary conditions are designed to do two things. The first is that the vertical displacement sends both beams into double curvature. The next is that the axial displacement produces higher curvature in the beams. There is some interface sliding in this problem, which when paired with the curvature for the given mesh discretizations at the two contacting surfaces, results in somewhat complex contact interface geometry. Similar to the previous problem in Section 11.2.2, the two beams have a tendency to "kick" up from one another in the regions of curvature, whereas contact occurs in a region near the point of inflection. To *force* more contact across the interface, an external pressure, (shown in Figure 39) is applied to sandwich the two beams together. The displacement boundary conditions are as follows: $\bar{u}_z = -1.0$m, $\bar{u}_x = -0.125$m and $\bar{\mathbf{t}} \cdot \mathbf{n} = 200$MPa is a Cauchy pressure and $\mathbf{n}$ is the unit noraml vector to the traction b.c. surface in the current configuration. Figures 40 and 41 show displacement contour plots of the two problem cases with conforming and non-conforming contact interface meshes, respectively. In both the conforming and non-conforming cases there is some amount of sliding between each beam. Additionally, there are regions of curvature near both supports and a point of inflection at the midspan of the assemblage. The applied external pressure on the top and bottom of the assemblage helps to force contact along the entire interface. Yet, similar to the discussion in Section 11.2.2, not all facet-overlaps experience zero gap compressive contact enforcement. Again, the subcycling converges to a kinematically possible solution wherein the active set of facet-pairs experiences some amount of distribution of zero-gap or separation constraint enforcement. This effect is mesh and problem dependent, but the important thing is that the subcycle procedure handles proper constraint

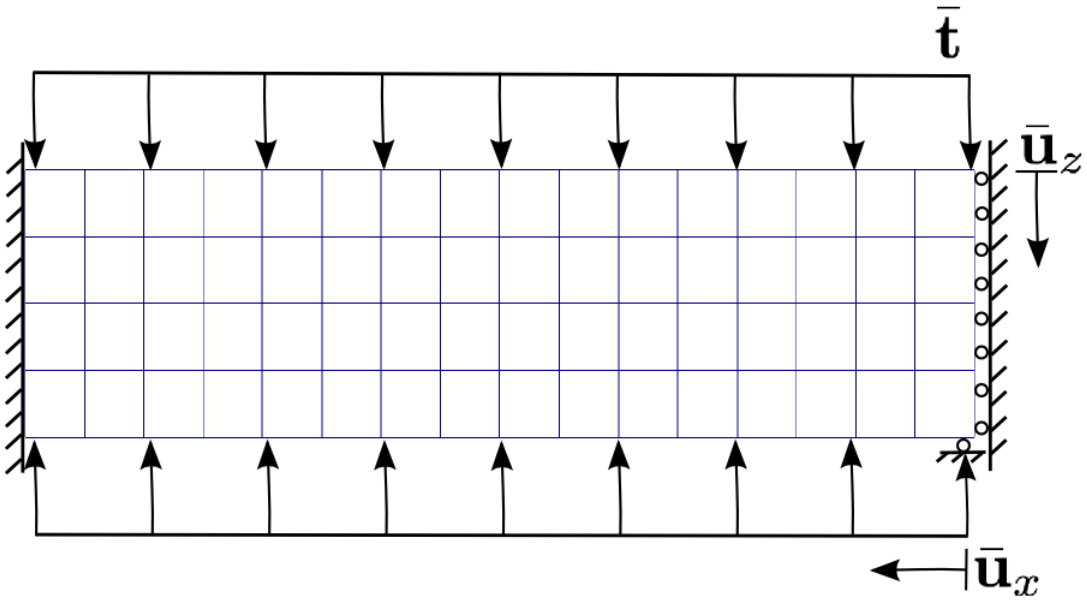

Figure 39: Diagram showing boundary conditions and support conditions for the stacked, fixed-fixed contacting beam problem.

enforcement without regard to either.

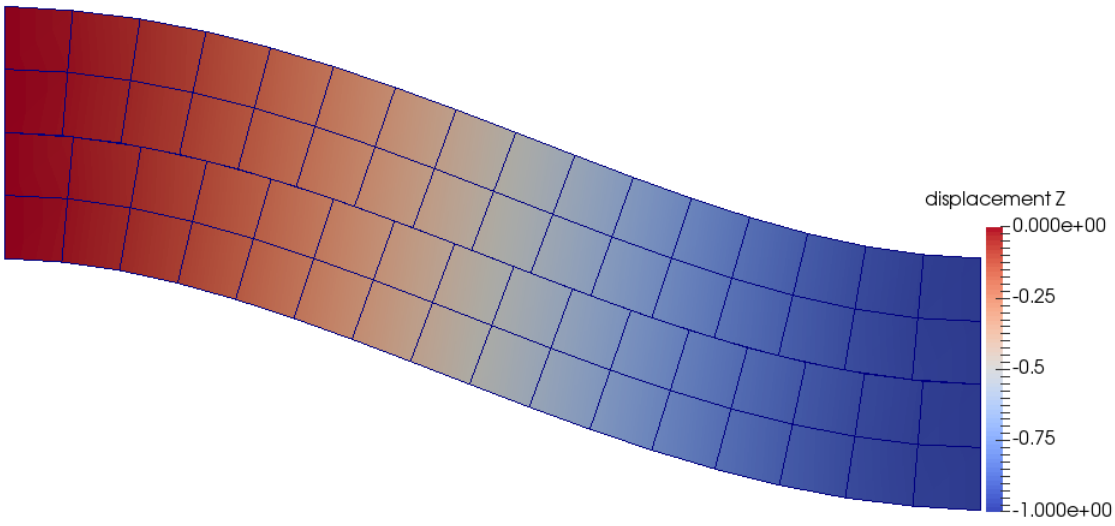

Figure 40: Final configuration z-displacement contour plot of the conforming stacked, fixed-fixed beam problem.

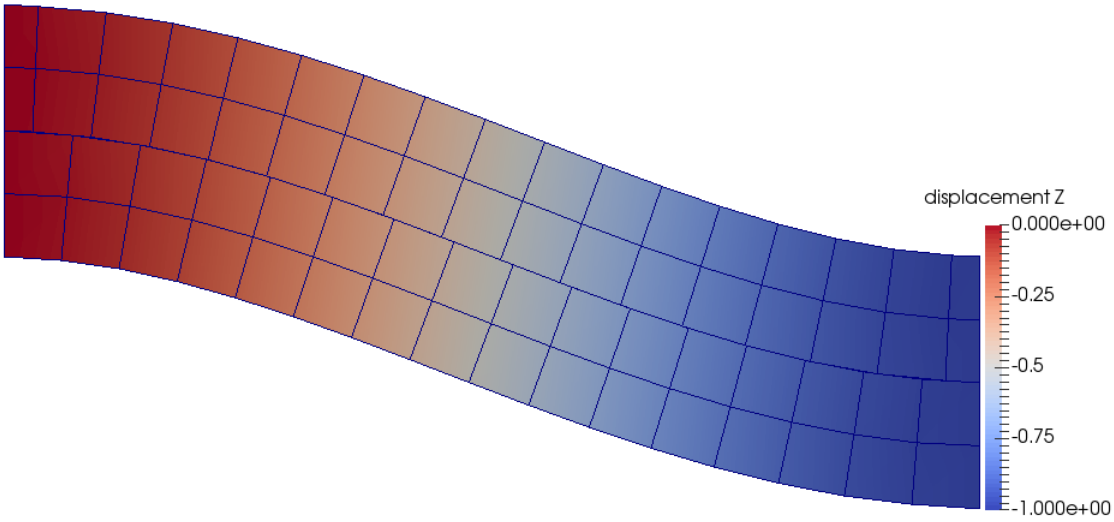

Figure 41: Final configuration z-displacement contour plot of the nonconforming stacked, fixed-fixed beam problem.

11.4 Contacting Cantilever Beams with a Perturbed Interface

The next set of numerical examples uses the hyperelastic stacked cantilever beam problem, but instead of having coplanar contact surfaces on each beam, a sinusoidally perturbed contact interface is introduced. The point in doing this is that when the cantilevers displace, the sliding causes *localized* contact interactions along the contact interface due to the perturbation.

11.4.1 Perturbed Interface Example 1

The first example uses sinusoidally perturbed contact surfaces on the top of the bottom cantilever and the bottom of the top cantilever. When the two beams are in their initial configuration, the two surfaces match exactly with zero gap across the interface. The initial configuration is shown in Figure 42.

A displacement boundary condition specified at the tip nodes of the top cantilever beam

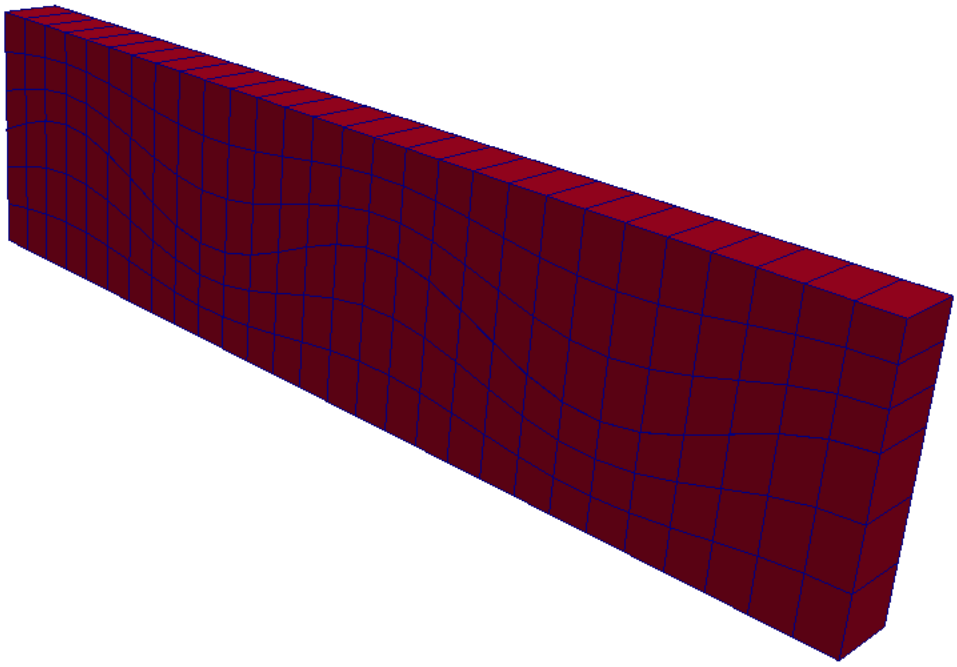

Figure 42: Initial configuration of the stacked cantilever beams with sinusoidally perturbed contacting surfaces.

initiates the deformation. The bottom cantilever displaces as a result of the contact along the interface. The relative sliding at the interface paired with the perturbed contact surface results in localized regions of contact at the slopes of the sinusoidal contact surfaces.

Figure 43 shows a displacement contour plot of the final configuration of the beam assemblage. Clearly the interaction results in regions of separation and regions of contact along the slopes of the sinusoidal interface. The regions of contact involve the near side of the sinusoidal waves at the top beam's contact surface sliding up the far side of the sinudoidal waves at the bottom beam's contact surface. Separation, then, occurs at the sides of the sinusoidal waves of each beam's contact surface opposite where contact occurs. As one surface slides "up" the other surface, the localized region of contact becomes smaller. Cleary, one can imagine that if this deformation process were to continue, then at some point only the crests of waves would be in contact. The problem was designed to create multiple regions of contact experiencing an evolution in the active contacting set of facet-pairs. In addition

to the displacement contour plot, Figure 44 shows a contour plot of the T_{33} stresses in the final configuration of the beam assemblage. One may observe increased compressive stresses local to the contacting portions of the interface.

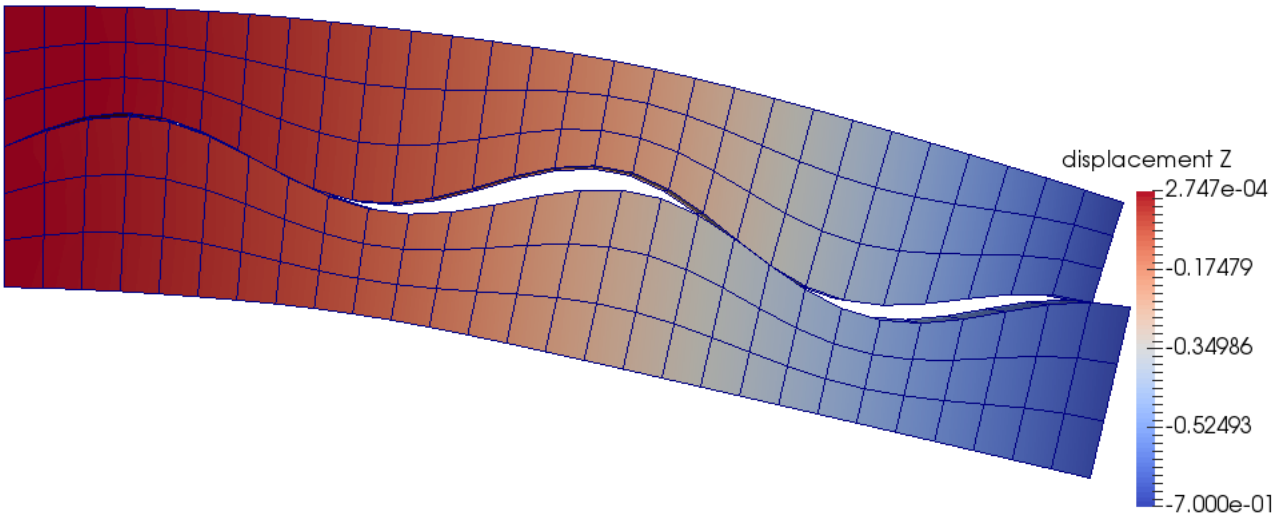

Figure 43: Final configuration z-displacement contour plot. Note the regions of contact and the regions of separation. Both such regions evolve through the deformation process as a result of interface sliding as the cantilevers displace vertically.

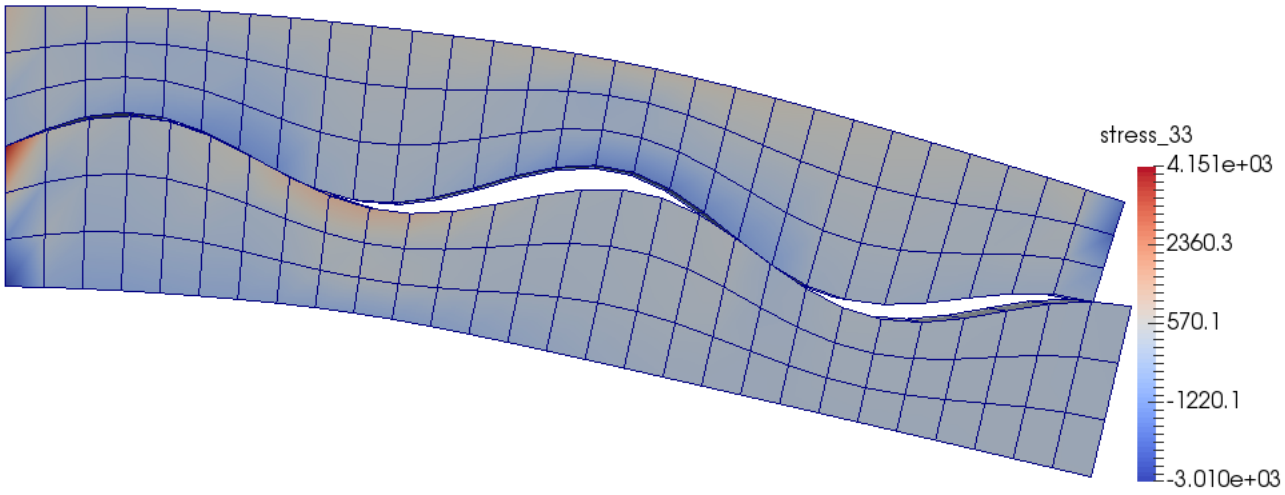

Figure 44: Final configuration T_{33} stress contour plot. Note, the contact stresses do not appear to be "equal-and-opposite." This is because the contact interface norma is not in the 3-direction and so continuity of the normal traction does not coincide with continuity of the T_{33} stress.

11.4.2 Perturbed Interface Example 2

This example features sinusoidally perturbed contact surfaces on each cantilever beam that geometrically match in the initial configuration, but unlike the previous example, the surface has a higher frequency and lower amplitude in the perturbation. When paired with the discretization of each contact surface, the result is a sawtooth like interface, the initial configuration of which is shown in Figure 45. Figure 46 shows a displacement contour plot

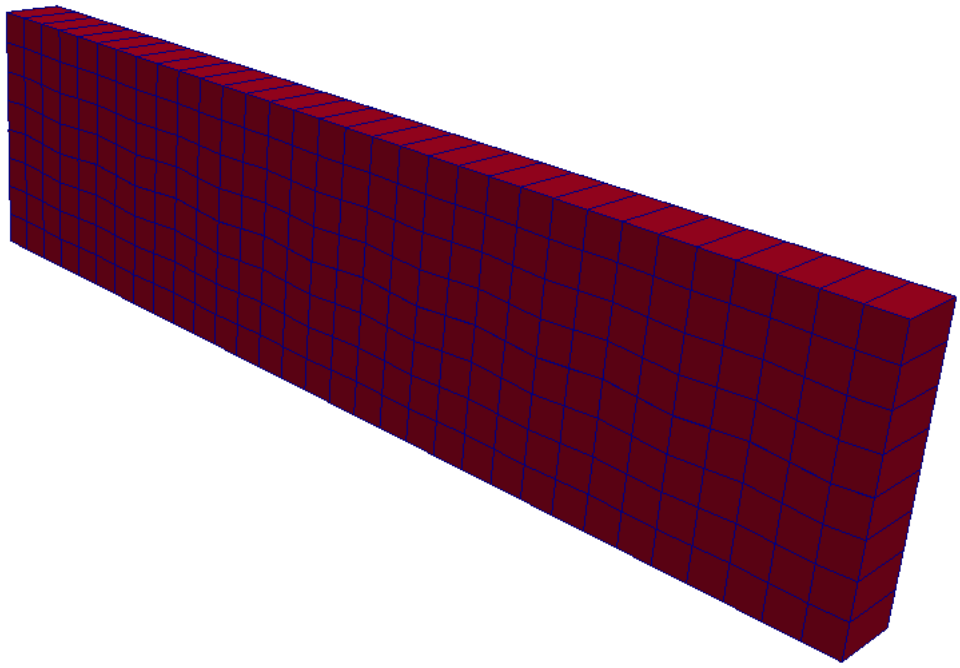

Figure 45: Initial configuration of the second example of stacked cantilever beams with sinusoidally perturbed contacting surfaces. Note that the perturbation in conjunction with the mesh density results in a sawtooth like interface.

of the final configuration of the beam assemblage. Similar to the stacked cantilever problem presented in Section 11.2.2, the top beam tends to "kick" up during the deformation process. As a result, the localized regions of contact are at the end of the beam and toward the support. How many facet-pairs in these regions are in contact is a matter of the mesh discretizations; however, there is separation through most of the contact interface due to this physical effect. Figure 47 shows a closeup of a region of separation along the contact interface. The smallest

separation gaps are such that these opposing facet-pairs may be included in the active set. The subcycling procedure is what determines whether contact enforcement will occur here. In general, the separation gaps are small and as sliding occurs, the contact interface evolves and the subcycling procedure must determine which facet-pairs are in compressive contact. The number of contact elements in compressive contact is very small compared to the size of the active set. This problem shows how the subcycle procedure can handle an over-populated active set and enforce contact in highly localized regions.

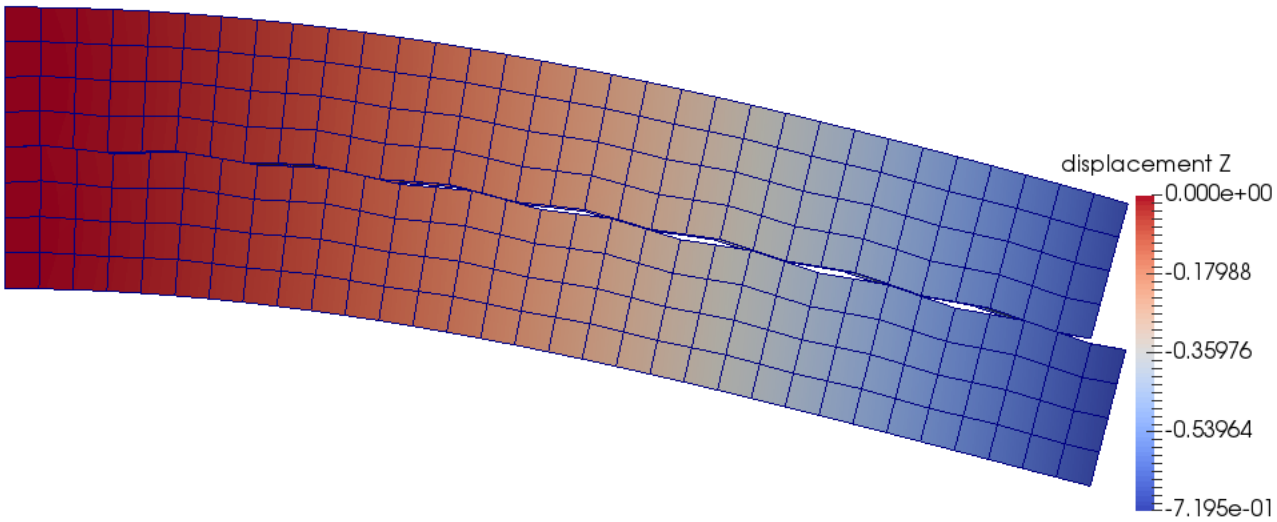

Figure 46: Final configuration z-displacement contour plot. Note that most of the interface is in separation, though geometrically within the active set tolerance that dictates which facet-pairs are included as contact elements.

Lastly, Figure 48 shows contour plots of the T_{33} stress at the final configuration of the beams. The contact is localized at the end and toward the support over very few facet-pairs that are actually in compressive contact due to the top beam kicking up from the bottom beam. With the separation shown in Figure 47 in mind, notice that the T_{33} stress in the top row of elements on the bottom beam is tensile. This is a point that will bear some significance in the next example.

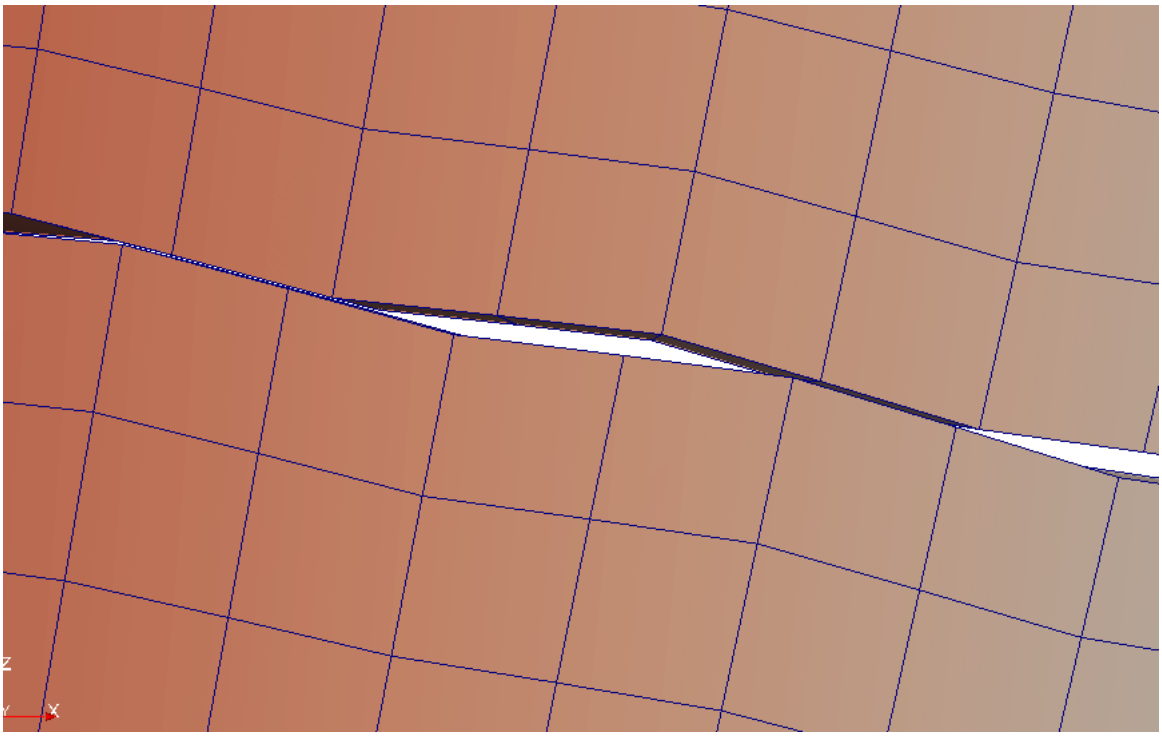

Figure 47: Closeup view showing two regions of the perturbed interface included in the active set due to their geometric proximity, yet exhibiting separation and no contact enforcement.

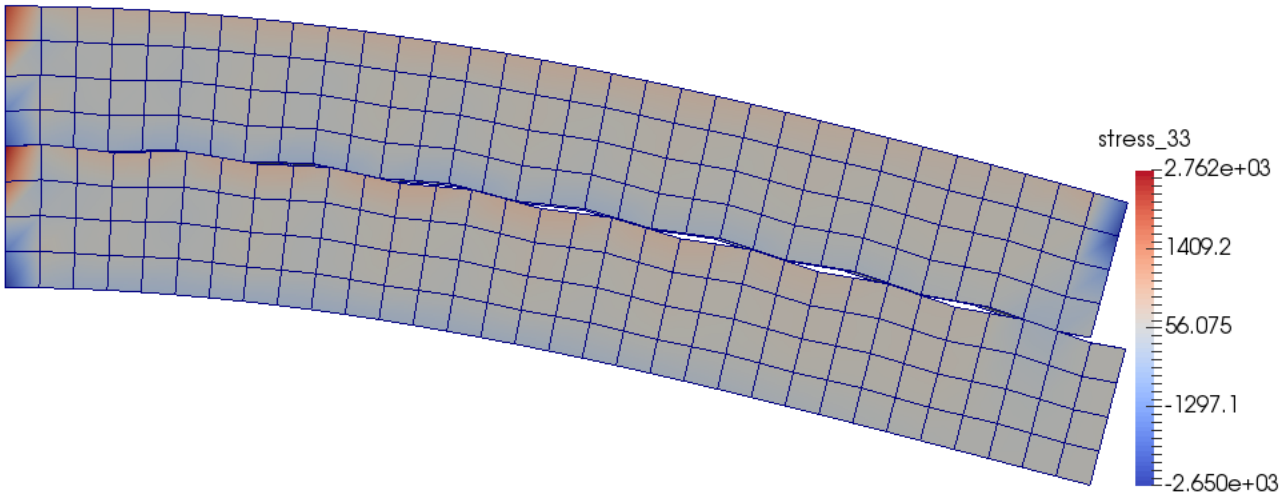

Figure 48: Final configuration T_{33} contour plot.

11.4.3 Perturbed Interface Example 3

This problem is the same as the previous problem presented in Section 11.4.2, except an external pressure is applied to the top and bottom of the beam assemblage in order to force contact across more of the interface. This applied Cauchy pressure need not be large; it is taken as $\bar{\mathbf{t}} \cdot \mathbf{n}^{(i)} = 50\text{MPa}$, where $\bar{\mathbf{t}}$ is the prescribed Cauchy traction and $\mathbf{n}^{(i)}$ is the current configuration normal at the top (i = 1) and bottom (i = 2) surfaces of the beam assemblage as appropriate. The initial configuration is shown in Section 11.4.2 in Figure 45.

Figure 49 shows a displacement contour plot of the final configuration of the beams. The Cauchy pressure applied to the top and bottom of the assemblage forces more of the interface into compressive contact. Showing this, the portion of the interface shown closeup in Figure 50 is the same as that shown in Figure 47. Clearly, the facet-pairs closest to contacting in the latter figure are actually in compressive contact in the former. Furthermore, as the beams deform, sliding occurs such that one contact facet slides across at most two opposing contact facets. The sawtooth interface is such that this sliding results in certain facets on one surface, for example, coming into and out of contact, while other facets either stay in contact (e.g. facets toward the support, which experience very little sliding) or remain in separation once sliding has occured (e.g. facets near a sawtooth crest that separate upon the initiation of sliding). While the problem in Section 11.4.2 had to enforce contact in small localized regions over a very small number of facet overlaps, this problem has to enforce contact in a more dynamically evloving contact interface as just described. This tests the subcycle procedure's ability to handle a contact interface where multiple contacting regions experience facet-pairs that come into, remain, and/or come out of contact. This problem demonstrates such a dynamic interface. Lastly, Figure 51 shows a contour plot of the T_{33} stress. An important observation is that the T_{33} stress at the bottom row of elements on the top beam is compressive and the stress at the top row of elements on the bottom beam is tensile. Even at locations of contact on the bottom beam the T_{33} stress is tensile. To the

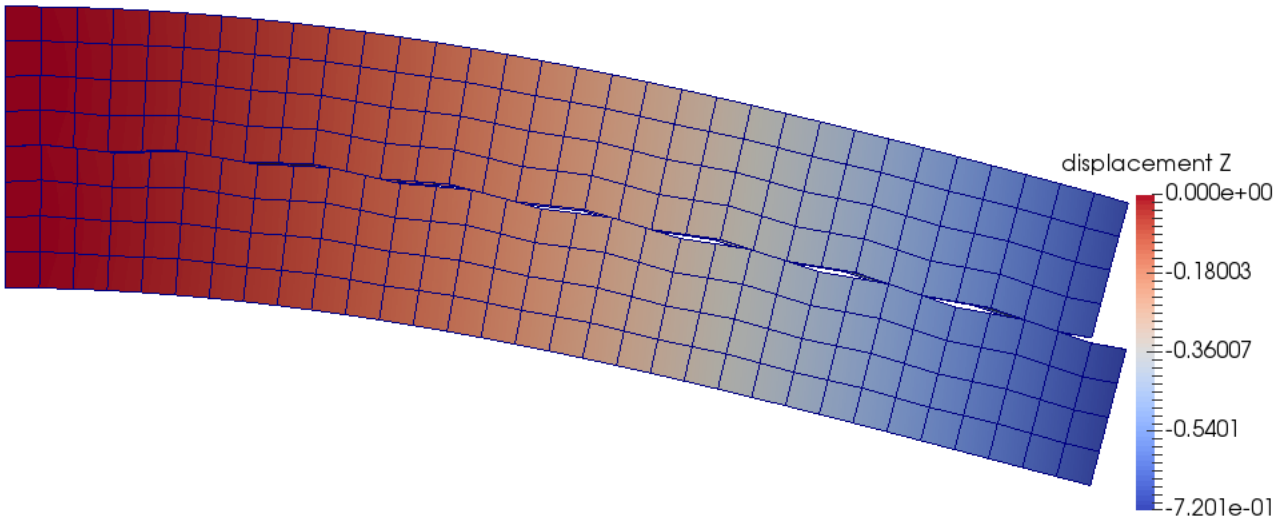

Figure 49: Final configuration z-displacement contour plot. Note that the externally applied pressure to the top and bottom surfaces of the beam assemblage force contact over more of the contact interface than is seen in the example presented in Section 11.4.2.

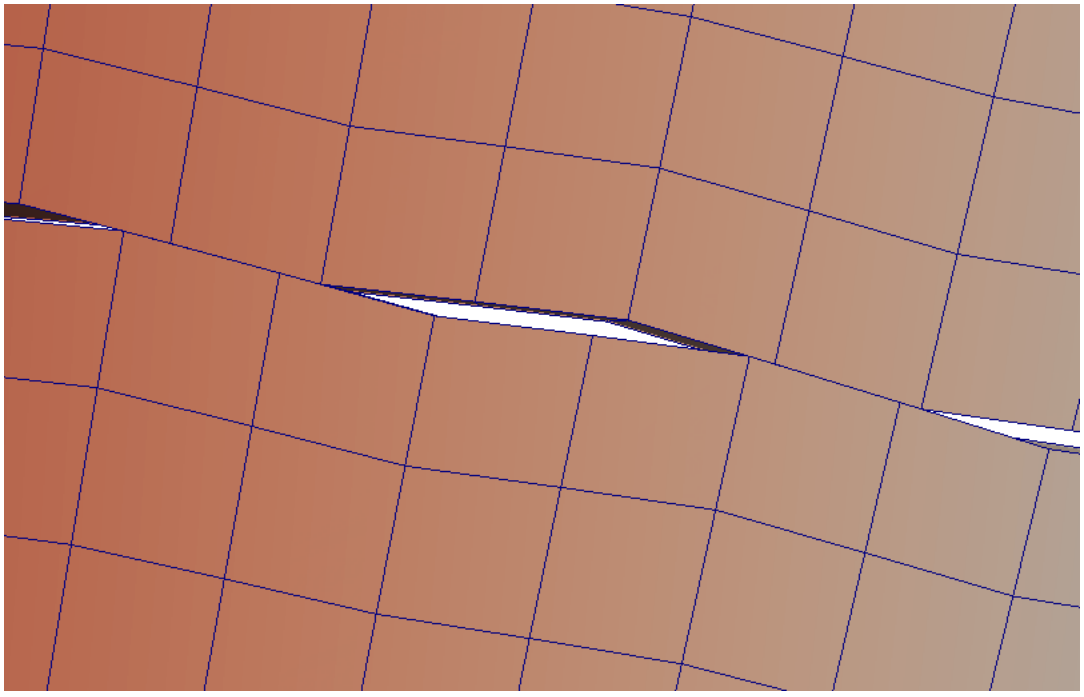

Figure 50: Closeup view of the same regions of the perturbed contact interface that were in separation in the problem presented in Section 11.4.2 that are now in contact as a result of the externally applied pressure.

contrary, however, the T_{33} stress at the locations of contact on the top beam are compressive. This is initially perplexing in that one would expect net compression at the elements whose facets form contact elements across the interface. In fact, the applied external pressure is intended to create contact across the whole interface, so observing tensile T_{33} stresses at elements on the bottom beam whose facets participate in contact elements took some further investigation.

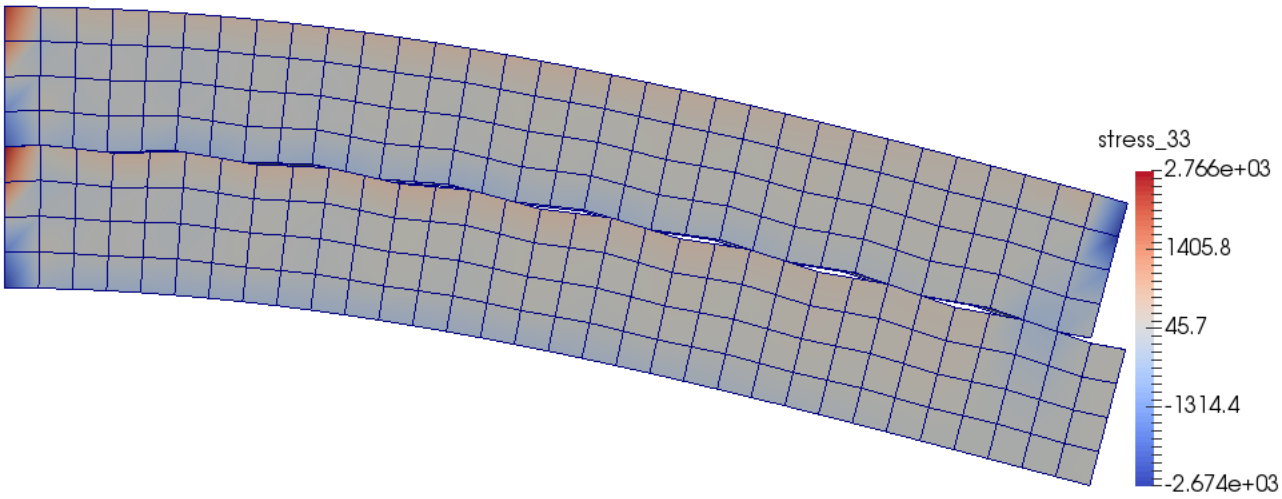

Figure 51: Final configuration T_{33} stress contour plot.

Figure 52 shows a single cantilever beam deformed to the same tip displacement as the problem under consideration using the same displacement boundary conditions at the tip nodes. This figure is a contour plot of the T_{33} stress, which shows that the top chord of the beam, which is in axial tension, experiences a tensile T_{33} stress. What this means is that the contact pressures in the stacked cantilever problem do not exert enough compressive pressure to result in net compression at the elements on the top chord of the bottom beam. Please note that, in contrast to this statement, the deformation is largely driven by contact at the end of the beams, which does in fact show net compression on both sides of that local interface.

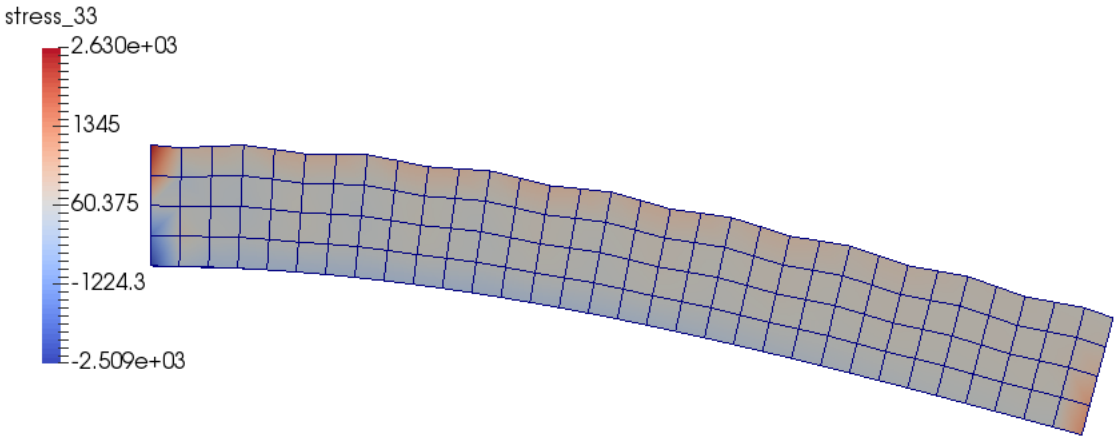

Figure 52: T_{33} stress contour plot of a single cantilever beam with the sinusoidally perturbed top surface. This figure is included to show that the T_{33} contact solution is in combination with the T_{33} bending stress solution.

11.5 Contacting Buckled Beams

This problem consists of two fixed-fixed, hyperelastic beams offset vertically. The material properties are the same for the Mooney-Rivlin material model as the stacked cantilever beam problems in Section 11.2. Each beam has a length of 4 and height of 0.5. They have a width of 0.25 and are separated by a vertical gap of 0.25. Each beam has two elements through the depth and one element through the thickness. The top and bottom nodes at the right hand side of each beam have prescribed displacement boundary conditions in the negative x-direction. The prescribed displacements are eccentric such that at a critical axial compression each beam buckles toward the other. The beam configuration including support conditions and boundary conditions is shown in Figure 53. The prescribed displacement denoted $\bar{u}_x + \Delta_x$ is a slightly larger specified displacement. Specifically, $\bar{u}_x = 0.25$m and $\Delta_x = 0.005$m. The eccentricity in the specified axial displacement at each beam creates a small moment such that as the beam displaces axially, it reaches a critical point when it then buckles. The top beam's mesh is different than the bottom beam's mesh such that when contact occurs between the buckled beams, the contact interface is composed of non-conforming meshes.

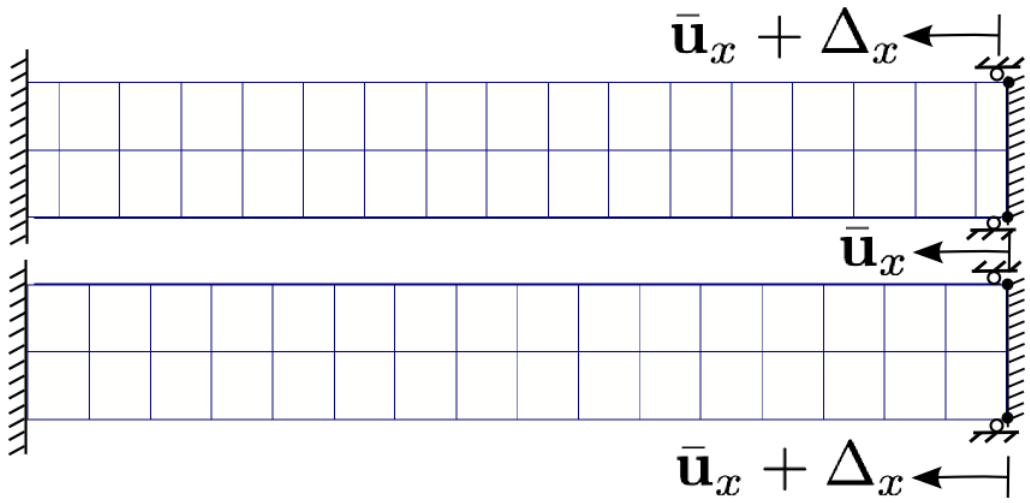

Figure 53: Diagram showing the boundary conditions and support conditions for the dual, fixed-fixed beam problem where buckling behavior is induced through displacement boundary conditions.

Figure 54 shows the displacement contour plots at various points in time during the simulation. Each snapshot, or "frame", in time may be referred to by the timestamp noted in the figure. One may observe the axial deformation up to $t_{15} = 0.375s$ whereas $t_{16} = 0.40s$ shows that buckling has initiated. As the beams displace toward each other, such as that shown in the $t_{18} = 0.45s$ frame, they eventually contact as shown in the $t_{19} = 0.475s$ frame. Finally, full contact for this problem occurs at $t_{20} = 0.50s$, which shows a slighter larger contact interface compared to the previous frame in the image.

This problem is a quasi-static problem. The time increments represent pseudo-time over which the simulation is conducted. The end of each timestep represents an equilibrium state. The timestep size in this problem is small enough that the buckling behavior of the two beams is adequately captured, including the first instance of contact. Again, the buckling is induced by the eccentric displacement boundary conditions at each beam.

To further demonstrate the effect the displacement boundary conditions have on the behavior of the two beams and the subsequent contact, we refer to Figure 55, which shows the

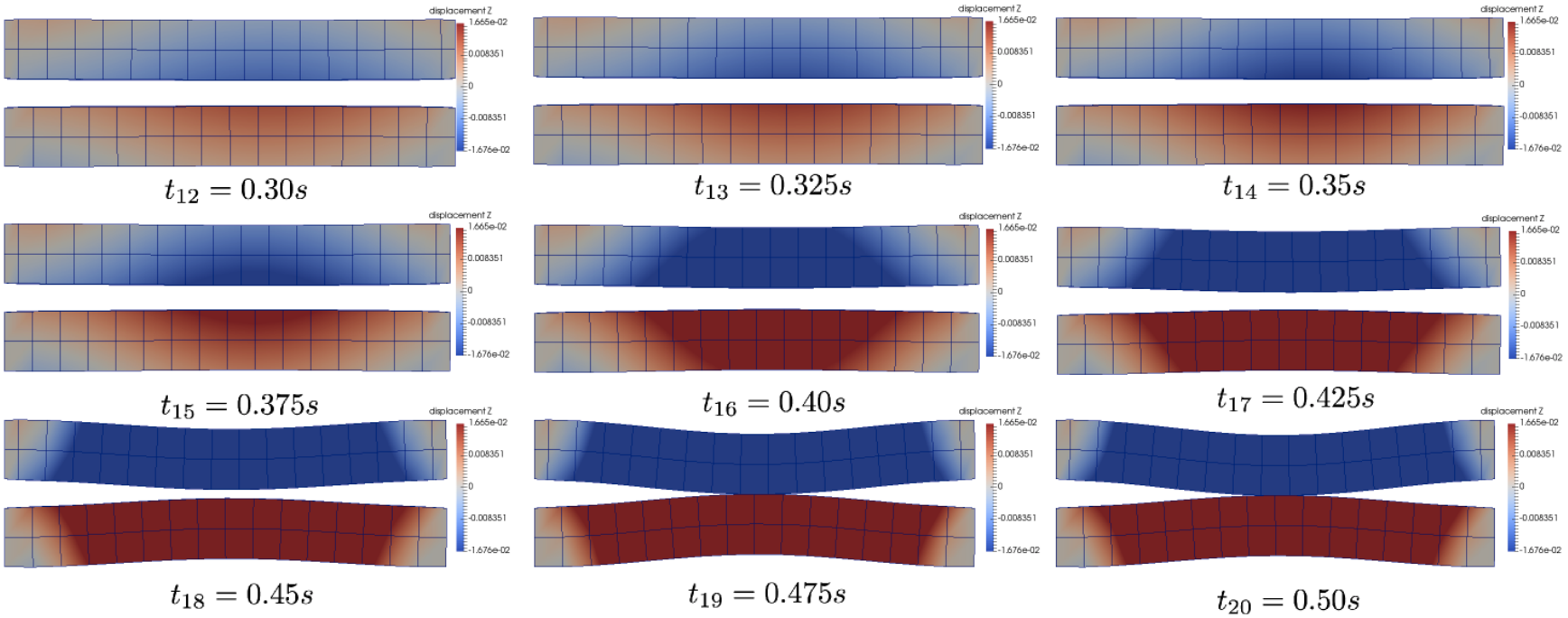

Figure 54: This sequence of plots shows the z-displacement solution contour plot at the noted points in time during the buckling simulation from continued axial displacement through to buckling and contact between the beams. The subscript, t_n, indicates the timestep number.

contacting state of the two beams. The pink node is a node of interest. Specifically, a plot of the vertical displacement vs. time at this node is shown in Figure 56. One can see that the vertical displacement increases at constant slope until buckling occurs where the slope increases and contact finally results. The contact position is then held throughout the rest of the simulation.

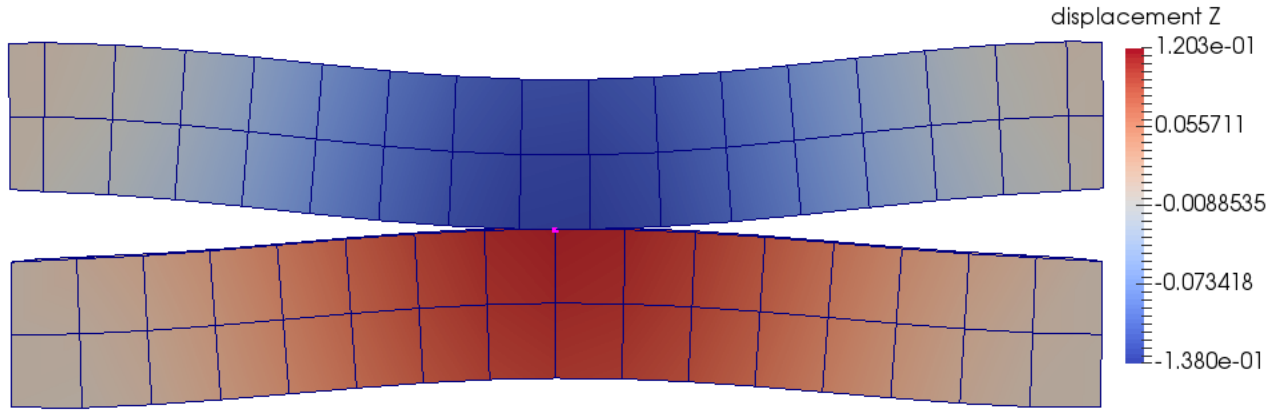

Figure 55: Final configuration z-displacement at contact between the two buckled beams.

The contact algorithm is designed to detect and enforce contact in problems where contact occurs abruptly, as demonstrated in this problem. A note on this point, however, is that this may require a smaller time step in order to capture the instance of contact and properly enforce it, which is largely an implicit finite element consideration.

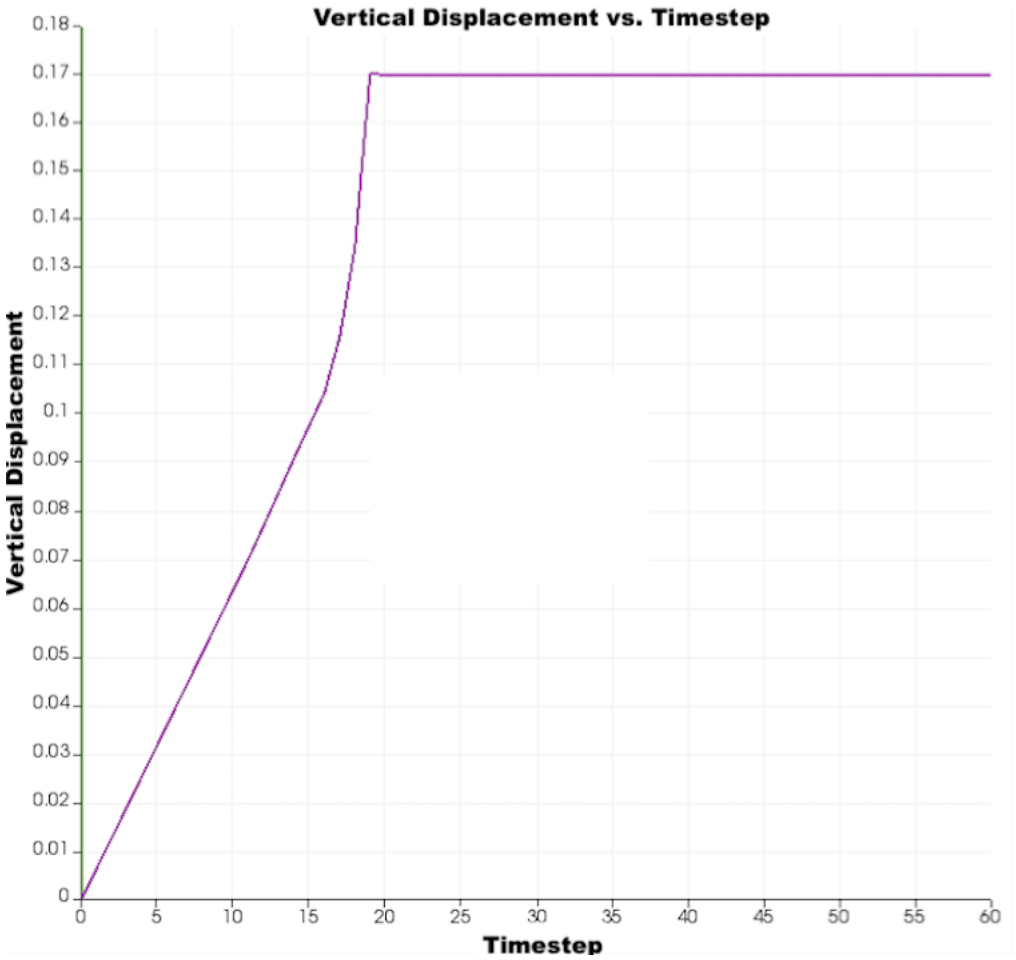

Figure 56: Diplacement (meters) vs. timestep plot at the center node on the top of the bottom beam indicated by a purple dot in Figure 55. Note the increase in slope as the beam buckles from approximately timestep 16 to approximately timestep 19, when contact occurs and the vertical displacement is held constant through the rest of the simulation.

11.6 Pressed Beam

This problem consists of two fixed-fixed beams separated by a vertical gap with the same beam dimensions as the previous problem presented in Section 11.5. An applied traction boundary condition is applied to the bottom side of the bottom beam, resulting in an upward displaced configuration exhibiting double curvature. In addition to the fixed-end constraints, the nodes of the bottom beam are constrained in the transverse direction. Each beam uses a hyperelastic material model with parameters set as for the stacked cantilever beam problems in Section 11.2. While still separated by a vertical gap, the top beam is then lowered via displacement boundary conditions specified in the negative z-direction at the nodes at each end of the beam, and *pressed* onto the deformed bottom beam. The top beam begins to assume the shape of the bottom beam through contact at the shared interface. While the top beam is pressed onto the bottom beam, the bottom beam also experiences subsequent deformation from the contact pressures applied in the negative z-direction. As a result, while the top beam is pressed onto the bottom beam, there is progressively more contact as the top beam begins to conform to the bottom beam geometry; yet, the contact interface evolves with the deformation of the bottom beam induced by contact with the top beam. Figure 57 shows the configuration of the two beams after the upward deformation of the bottom beam, but prior to contact with the top beam. The figure shows a contour plot of the T_{33} stresses, which are zero in the top beam and nonzero, as a result of bending, in the bottom beam. Specifically, the bottom surface of the bottom beam has an applied Cauchy traction of $\bar{\mathbf{t}} \cdot \mathbf{e}_3 = 3000\text{MPa}$ where $\mathbf{e}_3$ is the unit basis vector in the 3-direction.

As the bottom beam is fully deformed, the top beam is displaced downward and comes into contact with the bottom beam. Figure 58 shows a T_{33} contour plot (in MPa) with the initial contact between the two beams. As the top beam continues downward, the contact interface grows as the top beam deforms over the bottom beam. The bottom beam starts to displace while maintaining its double curvature as a result of the contact interaction. Figure 59 shows

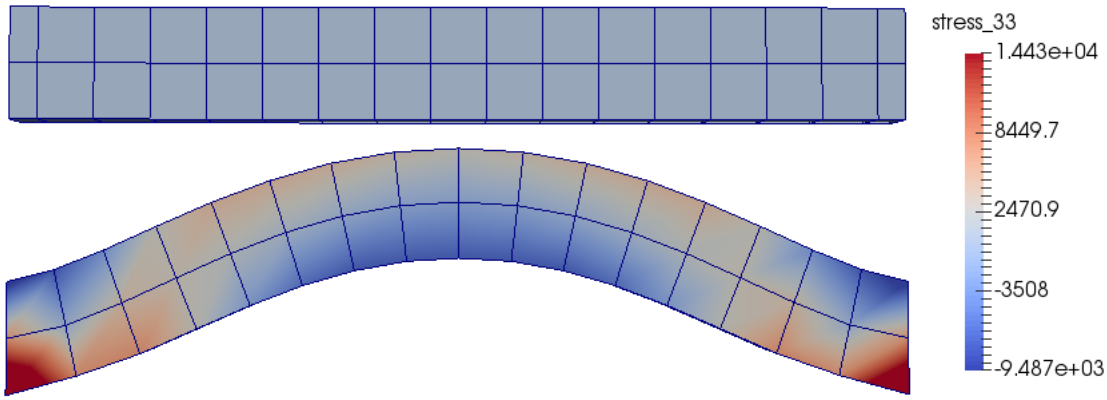

Figure 57: An intermediate, pre-contact configuration of the two beams in the pressed beam problem. This is a T_{33} stress contour plot. Note the stress distribution in the bottom beam due to the bending behavior.

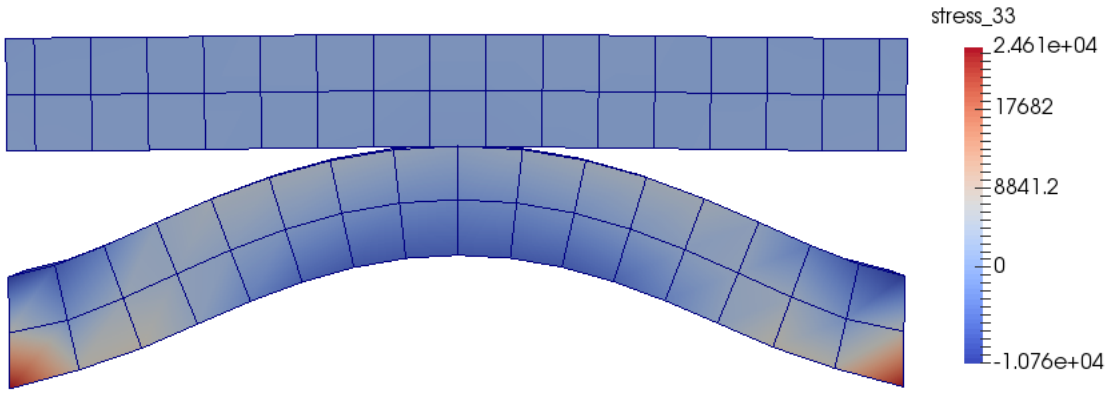

Figure 58: T_{33} stress contour plot showing the initial contact between the two beams.

a T_{33} contour plot showing such an intermediate configuration in the deformation process.

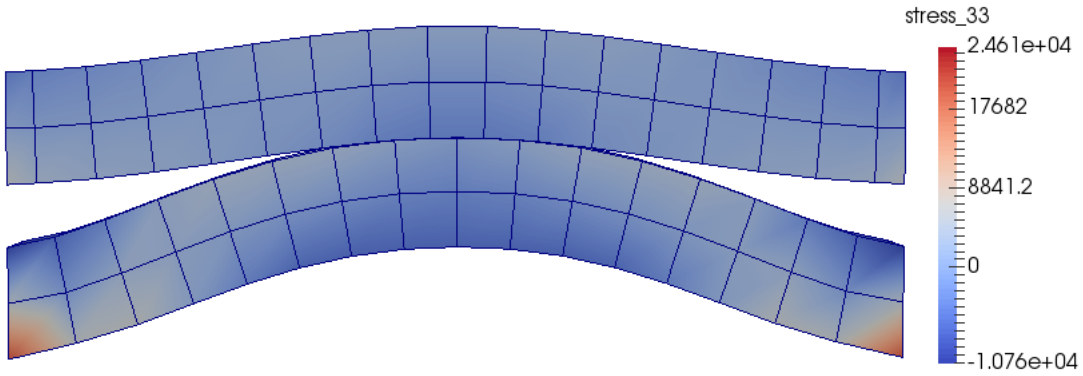

Figure 59: T_{33} stress contour plot showing an intermediate state of deformation where the top beam has come into greater contact with the bottom beam, begun to deform over the bottom beam while at the same time vertically displacing the bottom beam.

Finally, Figure 60 shows the end configuration of the two beams where the top is fully pressed onto and assumes the deformed shape of the bottom beam.

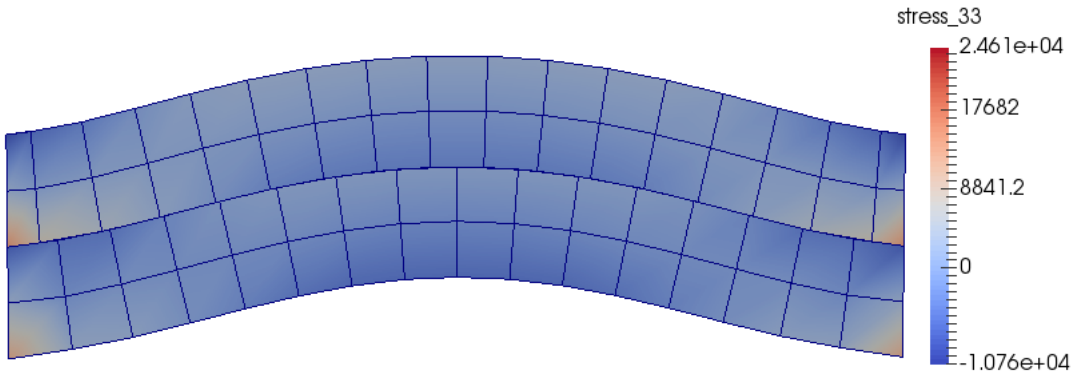

Figure 60: Final configuration T_{33} stress contour plot with the top beam in full contact pressed upon the bottom beam.

11.7 Brick Compression with Offset Perturbed Surfaces

This problem introduces two bricks with opposing, sinusoidally perturbed, contact surfaces. The top brick's bottom perturbed surface has a small phase shift such that when forced together, the two bricks will not match at the interface. The phase shift causes contact and subsequent axial deformation in the two bricks. The bottom brick is fixed at the bottom nodes in the transverse and vertical directions and nodes on the far left hand side are fully fixed, which resists the axial forces induced by longitudinal deformation. The top nodes of the top brick are restrained in the transverse direction and have a specified vertical displacement boundary condition, which brings the two bricks together. Simlar to the bottom brick, the nodes at the far right hand side of the top brick are fully restrained so as to provide a fixed support to resist logitudinal deformation caused by the contact interaction. The initial configuration of the two bricks is shown in Figure 61. This problem uses a hyperelastic material model with parameters set forth in Section 11.2.

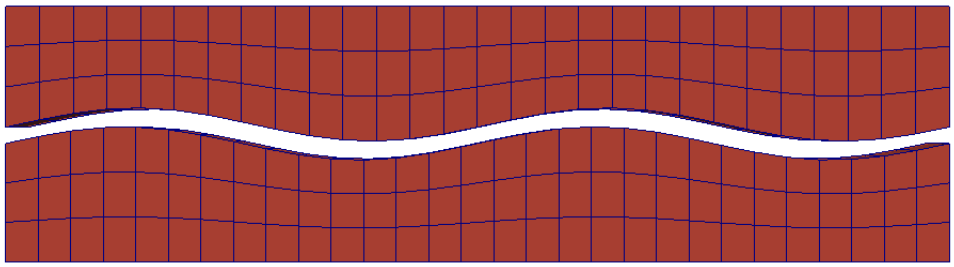

Figure 61: Initial configuration of two bricks with sinusoidally perturbed contact surfaces.

The displacement boundary conditions at the nodes on the top surface of the top brick bring the top brick into contact with the bottom brick. The top brick compresses the bottom brick and the phase shift results in initial localized contact regions on the left hand side, nonzero slopes of the sine waves. A T_{33} stress contour plot (in MPa) shows this initial contact in Figure 62. As the top brick continues to compress the bottom brick, the two bricks elongate longitudinally and slide into complete contact across the interface. Figure 63 shows a T_{33}

contour plot of the final configuration with contact across the entire interface between the perturbed surfaces. Note the axial elongation that has occurred, allowing the surfaces to effectively slide into phase and match with one another.

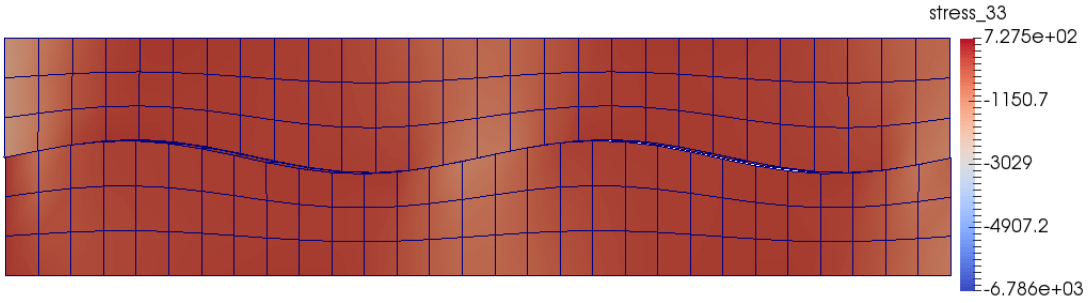

Figure 62: T_{33} stress contour plot at initial contact between the two bricks.

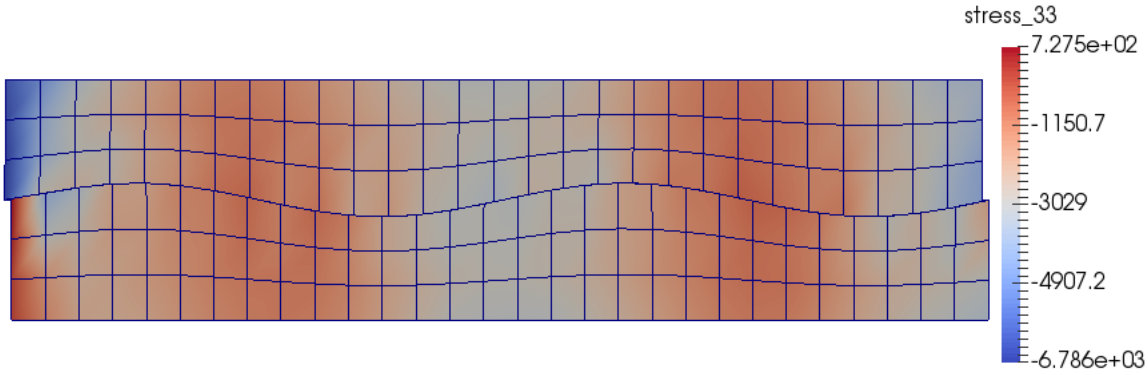

Figure 63: Final configuration T_{33} stress contour plot. Note the contact interaction between the out-of-phase contacting surfaces results in elongation of the bricks in the x-direction in order for the two perturbed surfaces to mate.

11.8 Sliding Brick with Perturbed Interface and Contacting Block

This problem includes a brick with a sinusoidally perturbed interface and an opposing contact block. Both use a hypoelastic material model with the following properties, E = 1.E6MPa, $\nu = 0.3$. The initial configuration of the two is shown in Figure 64. The perturbed brick is restrained in the transverse and vertical directions at the bottom nodes, while diplacement boundary conditions specified at the far right hand side end nodes translates the brick to the left in the negative x-direction. As the brick begins to translate to the left, the contacting block has lowered vertically to a fixed position and comes in and out of contact with the brick as the sinusoidal surface passes below.

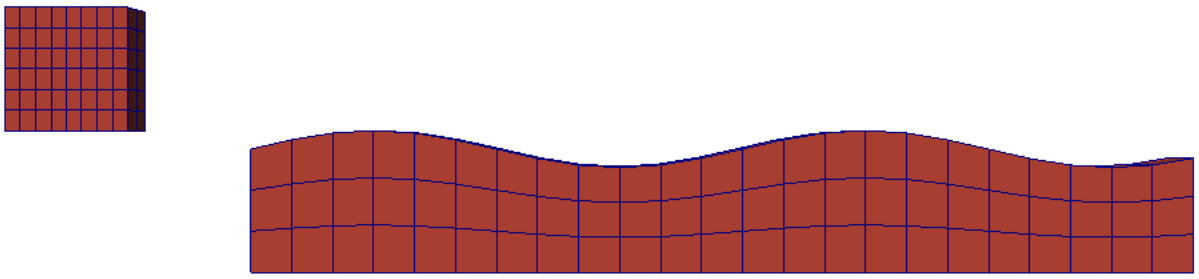

Figure 64: Initial configuration of the bottom brick with a sinusoidally perturbed contact surface and the top contacting block.

Figure 65 is a T_{33} contour plot (in MPa) showing the contact stresses at initial contact between the brick and contacting block. Figure 66 and Figure 67 are similar T_{33} contour plots showing two subsequent positions of the contacting block further along the first crest of the perturbed surface. As the brick continues to translate to the left, the contacting block eventually comes out of contact with the first crest as it hovers above the first trough of the sinusoidally perturbed surface. This is shown in Figure 68. Again, as the brick translates to the left, the contacting block comes once again into contact with the brick at the second crest of the perturbed surface. This is shown in a T_{33} contour plot in Figure 69.

This problem demonstrates the algorithm's ability to handle the contacting block coming into and out of contact along a perturbed, non-conforming surface. In fact, as the block comes into contact with the brick at one of the two crests on the brick's surface, the block must compress to the point that the entire bottom surface is in contact with the brick while deformed accordingly. As the brick continues to translate, the block returns to its original configuration while sliding out of contact with the brick.

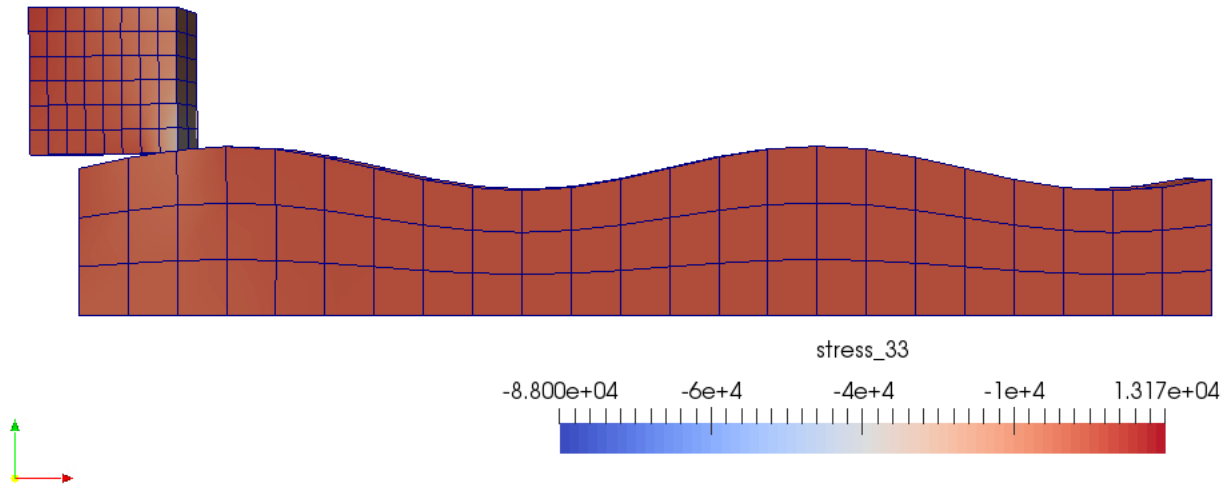

Figure 65: T_{33} stress contour plot at initial contact between the sliding brick and the contacting block.

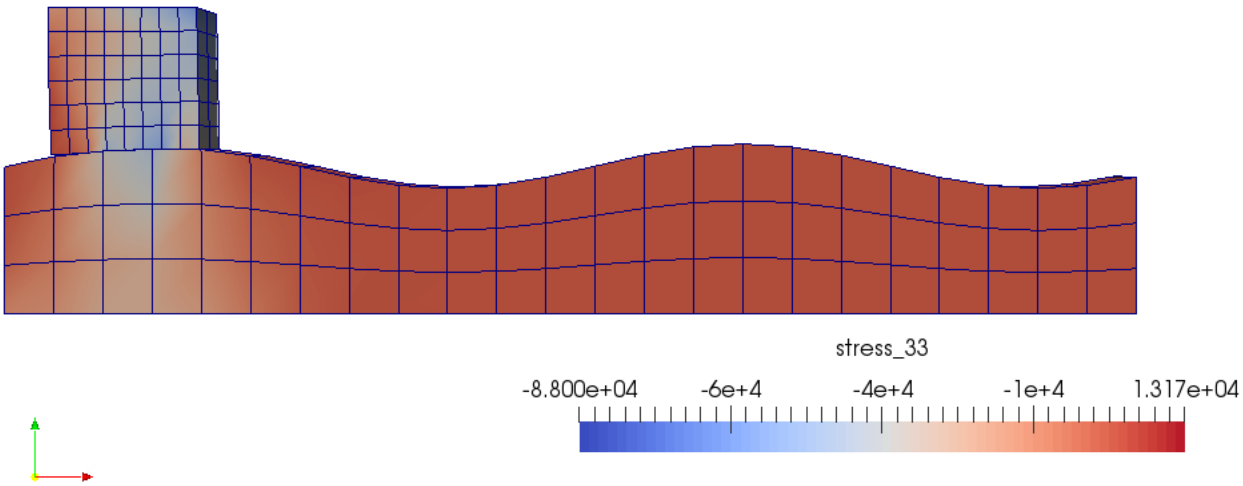

Figure 66: T_{33} stress contour plot showing the contacting block in complete contact with the crest of the first wave on the brick's sinusoidally perturbed surface.

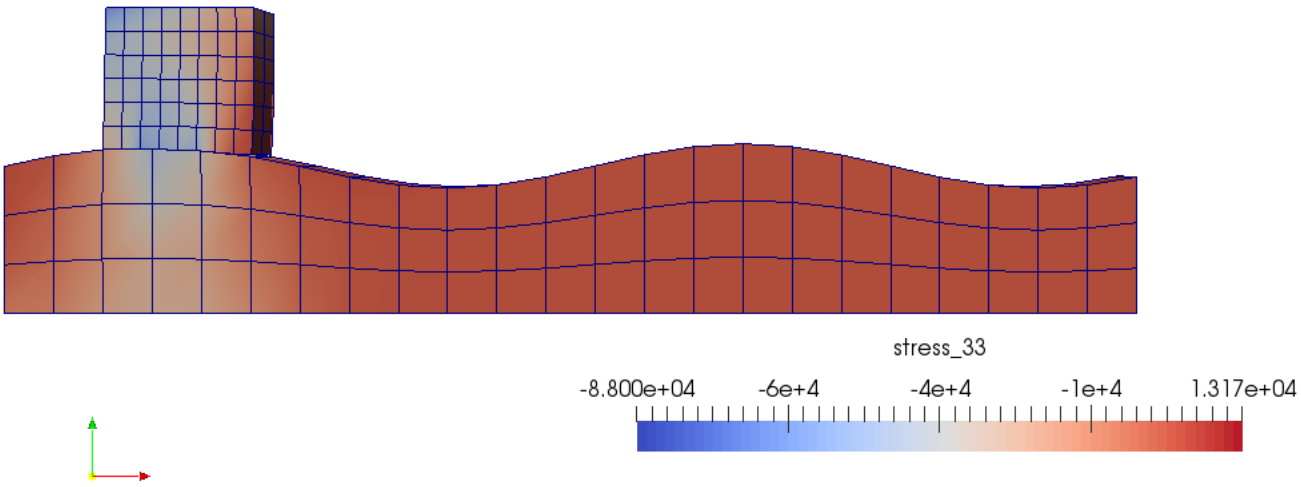

Figure 67: T_{33} stress contour plot as the contacting block is coming out of contact with the crest of the first wave on the brick's sinusoidally perturbed surface.

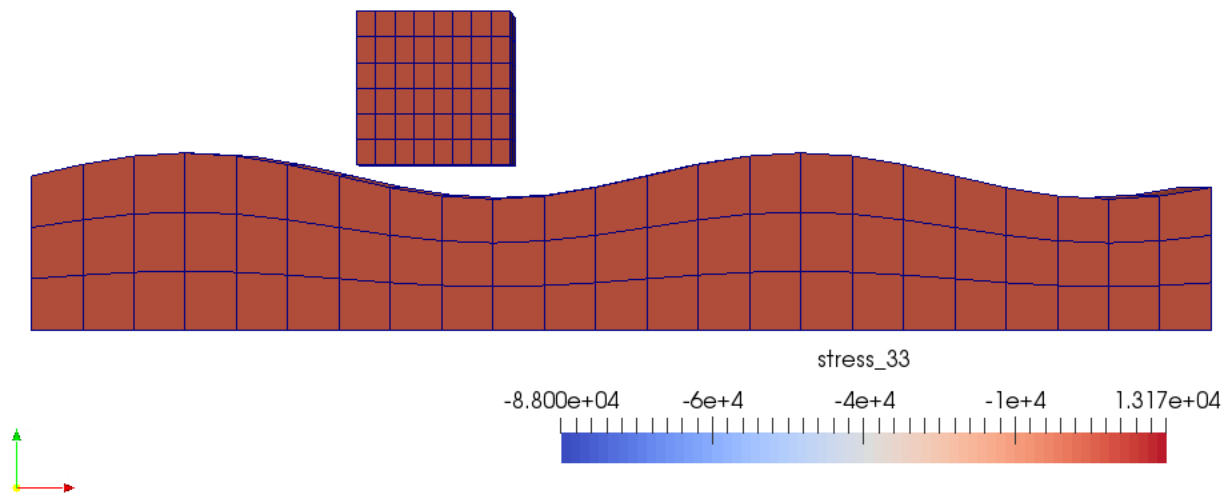

Figure 68: T_{33} stress contour plot showing no contact interaction as the contacting block rests above a trough on the brick's perturbed surface.

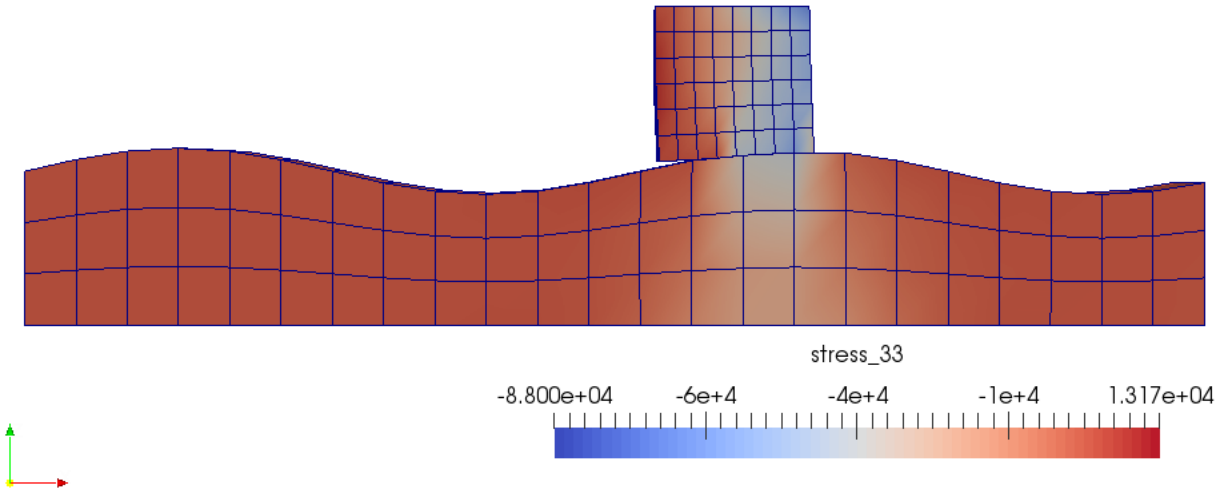

Figure 69: T_{33} stress contour plot as the contacting block comes into contact with the second crest in the brick's sinusoidally perturbed surface.

11.9 Sliding Brick-to-Brick with Perturbed Surfaces

This problem consists of two hypoelastic blocks with identical and opposing perturbed contacting surfaces sliding one over the other. Their material properties are the same as in the previous problem in Section 11.8. Initially, the two bricks are separated as shown in Figure 70 and while the bottom block remains stationary, restrained at the bottom nodes, the top block slides in the x-direction over the bottom block. This problem is designed such that when the two sinusoidally perturbed surfaces are out of phase then contact will occur, and when they are in phase, no contact, but rather a perfectly matching, zero gap interface will occur.

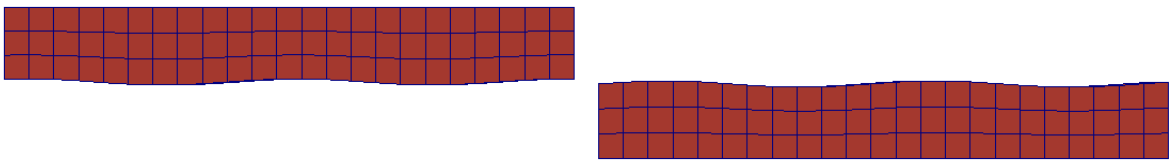

Figure 70: Initial configuration of the two bricks showing the sinusoidally perturbed contact surfaces.

Figure 71 shows a T_{33} contour plot of the initial contact between the two bricks. As the two slide over each other, the opposing crests on each perturbed surface contact and compress in order to allow the top brick to continue translating in the x-direction.

Figure 72 shows a similar T_{33} stress contour plot where the initial contacting crests are in full contact. Further translation of top brick will result in the two bricks coming progressively out of contact until the two perturbed surfaces are in phase with one another. That latter instance is shown in Figure 73 where there is zero contact (and therefore zero T_{33} stress) between the two bricks.

Further translation will result in two crests contacting, as shown in the T_{33} contour plot in

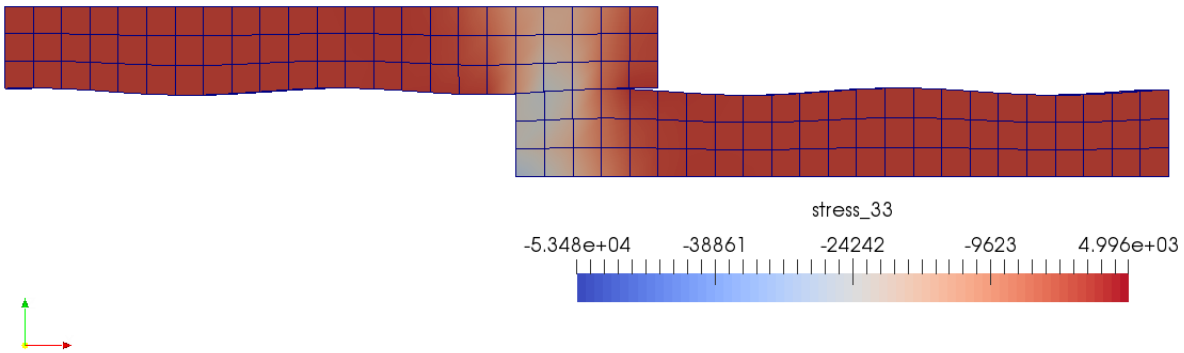

Figure 71: T_{33} stress contour plot at the first instance of contact.

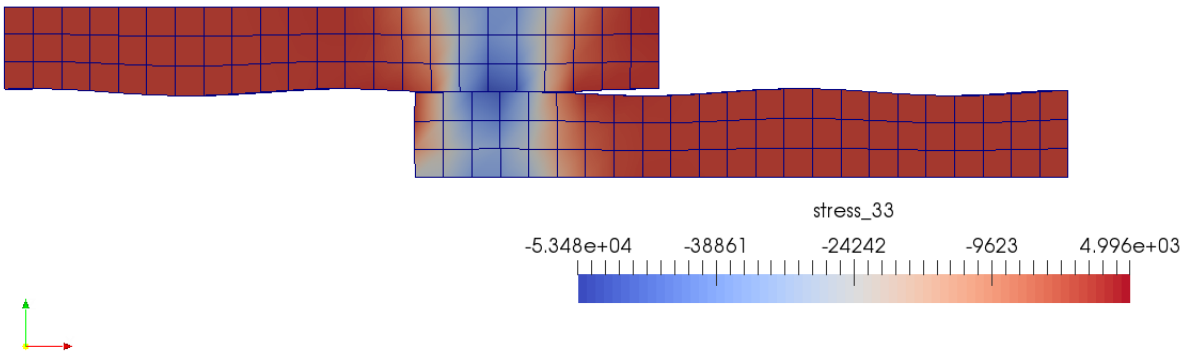

Figure 72: T_{33} contour plot showing the contact stresses as the first two crests in each contact surface slide progressively furthere into contact.

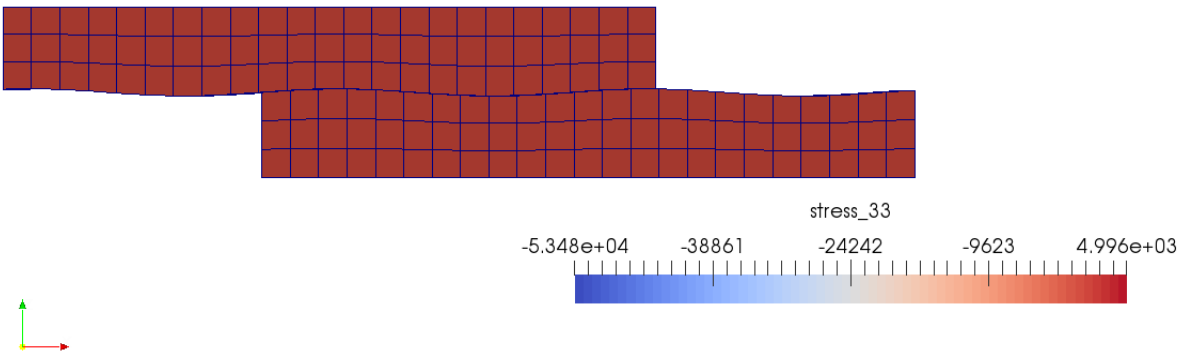

Figure 73: T_{33} stress contour plot showing no contact as the two perturbed surfaces are in phase with one another.

Figure 74, with subsequent phase matching at the interface as shown in Figure 75. This problem is similar to the problem presented in Section 11.8, but this problem has a more complex interface that must come into and out of contact where the in-phase, out-of-contact is a zero gap, perfectly matched interface.

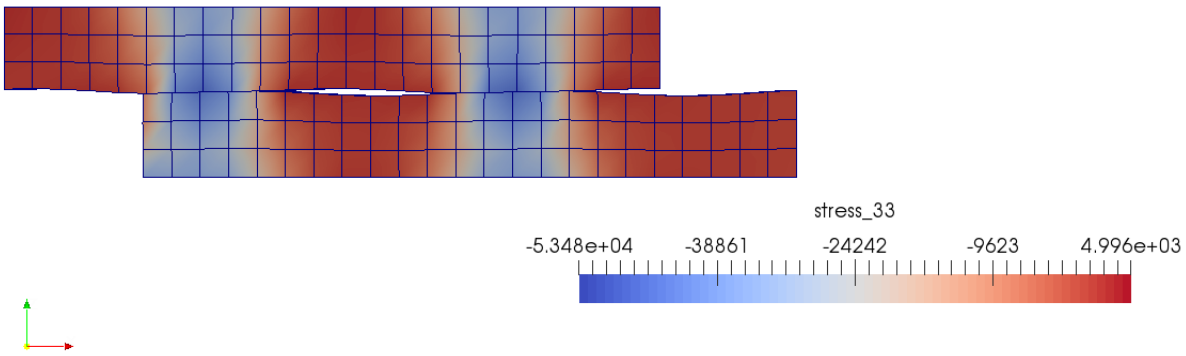

Figure 74: T_{33} stress contour plot with two localized regions of contact. As the top brick continues to translate, two crests in the perturbed surfaces come into contact.

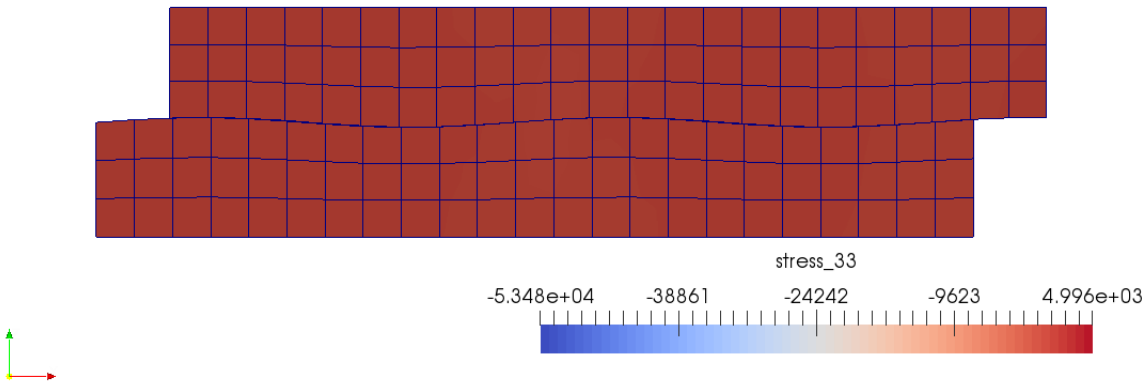

Figure 75: T_{33} stress contour plot showing no contact as the two surfaces are in phase with a larger zero gap interface.

11.10 Sliding Brick-to-Beam with Perturbed Interface and Cantilever Action

This is a hypoelastic problem composed of a brick that slides under a cantilever beam, each with opposing sinusoidally perturbed contact surfaces. The material properties are the same as those presented for the problem in Section 11.8. The bottom brick, with transversely and vertically restrained bottom nodes, translates in the x-direction sliding underneath the bottom surface of the top cantilever beam. When sliding underneath, the sinusoidally perturbed contact surfaces go in and out of phase. As a result, when out of phase, the contact at the surface crests causes the cantilever to deform upward, whereas when in phase, the two surfaces match exactly with a zero gap interface. The initial configuration of the brick and beam is shown in Figure 76. For clarity, Figure 77 shows a closer perspective view of the bottom brick so the reader may more easily see the sinusoidally perturbed top surface.

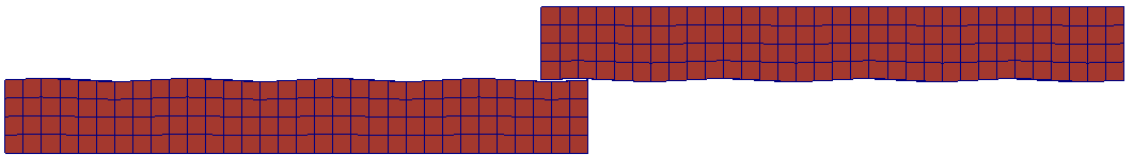

Figure 76: Initial configuration of the bottom brick and top cantilever, both with sinusoidally perturbed contact surfaces.

As the first two full crests of the perturbed surfaces come into contact, the cantilever displaces upward. This deformation pattern is shown in the T_{33} contour plot in Figure 78. As the two surfaces come into phase with one another as the bottom brick continues in x-direction translation then there is an instant of zero gap across the entire contacting interface. This is shown in Figure 79.

Figure 80 demonstrates the fact that the contact that results in the upward displacement

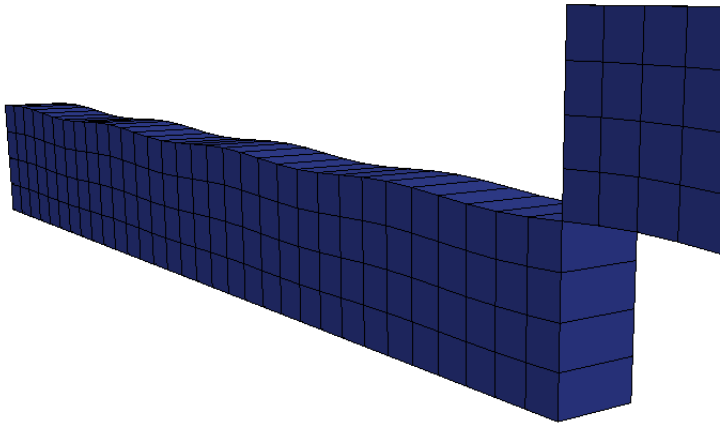

Figure 77: Closeup view showing the sinusoidally perturbed top surface of the bottom brick. The bottom surface of the top beam is an identically perturbed surface such that when the beam rests perfectly over the brick, the two surfaces mate with a zero gap interface.

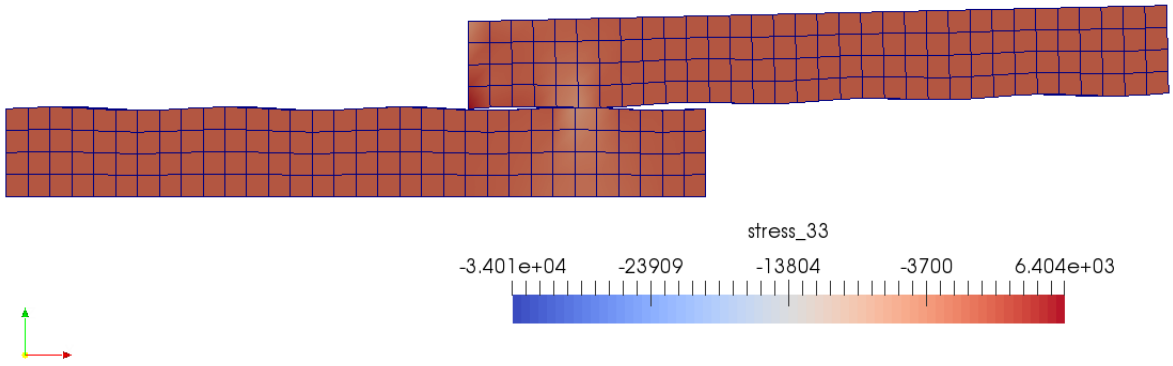

Figure 78: T_{33} stress contour plot showing the first instance of contact and the resulting vertical displacement of the top cantilever beam.

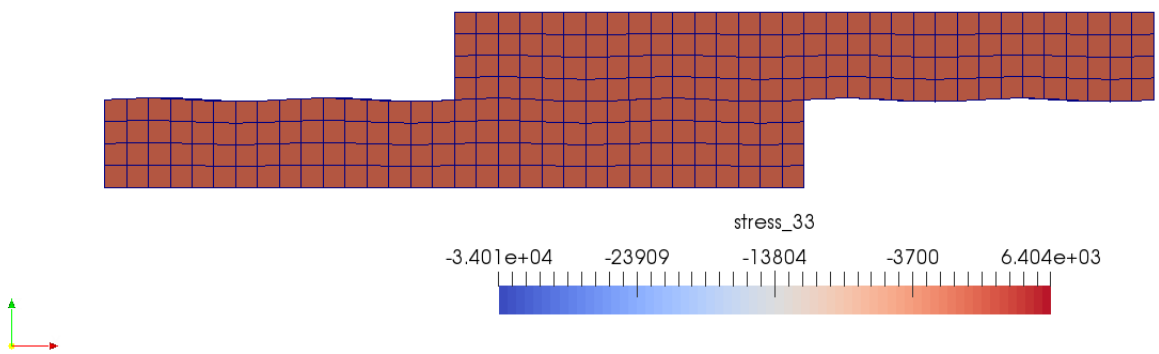

Figure 79: T_{33} stress contour plot showing the two contacting surfaces in phase with a zero gap interface and zero contact.

of the cantilever beam is caused from each crest on the contact surface of the bottom brick coming into contact with the first full crest on the contact surface of the cantilever beam. As the brick continues to translate, however, an increasing amount of the two surfaces "mate" in a zero gap interface as the cantilever returns to its initial position as shown in Figure 81 where the brick and beam mate across the entire interface.

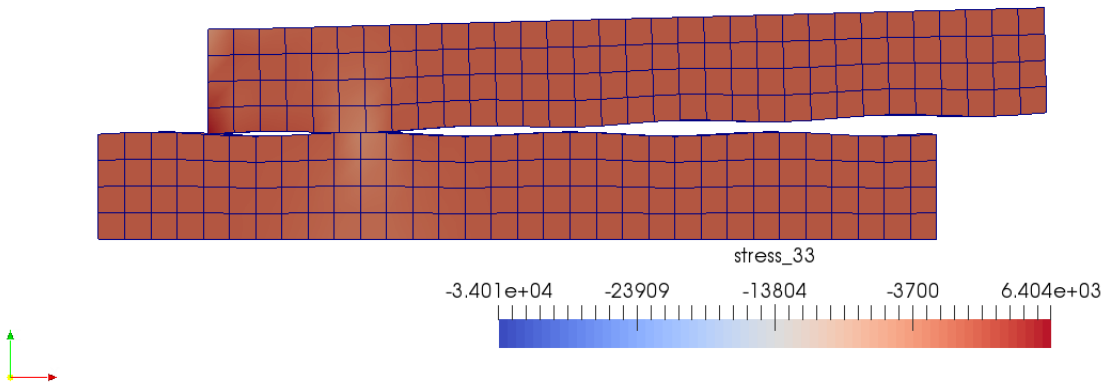

Figure 80: T_{33} stress contour plot showing a later state in the translation of the bottom brick under the top beam. The contact still occurs at the first crest on the left hand side of the cantilever beam's perturbed surface with a resulting vertical displacement.

The up-and-down motion obviously repeats for as long as the brick is sliding underneath

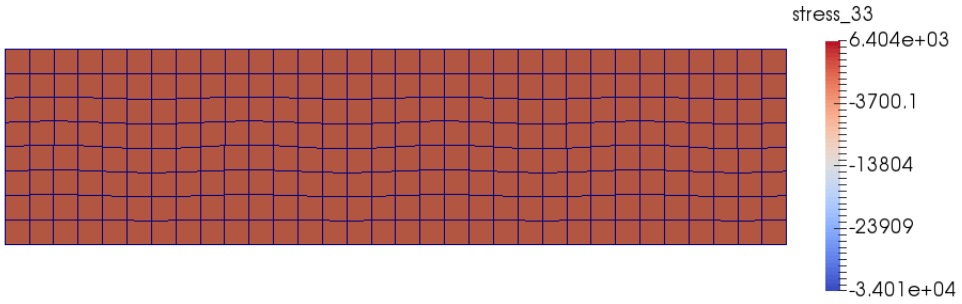

Figure 81: T_{33} stress contour plot showing a perfect, in phase, zero gap interface between the brick and the beam.

the cantilever beam, but as the last crest on the left hand side of the brick's perturbed surface comes into contact with the cantilever beam, the contact begins to occur at each of the top beam's crests one after the other approaching the end of the beam. Figure 82 shows contact where the left hand side of the bottom brick has passed the mid-length of the cantilever beam. One can see that the tip displacement of the cantilever beam is clearly smaller than in the initial contact interaction. In fact, Figure 83 shows a displacement vs. timestep plot at the end of the cantilever. First off, the unsmooth behavior in the first vertical displacement excursion is primarily due to the fact that we have an upward trending, partial sine wave at the right-end of the brick that first comes into contact with the cantilever beam. Mesh refinement or a slight alteration of the perturbed surface such that the sine wave trends downward at the right-end of the brick would result in a smoother initial contact enforcement. That aside, the plot shows a smooth oscillation in the tip displacement as the brick slides underneath. Note that the amplitude of the tip displacement begins to decrease as the bottom brick is fully underneath the cantilever and continues to translate to the right in the x-direction, as previously noted.

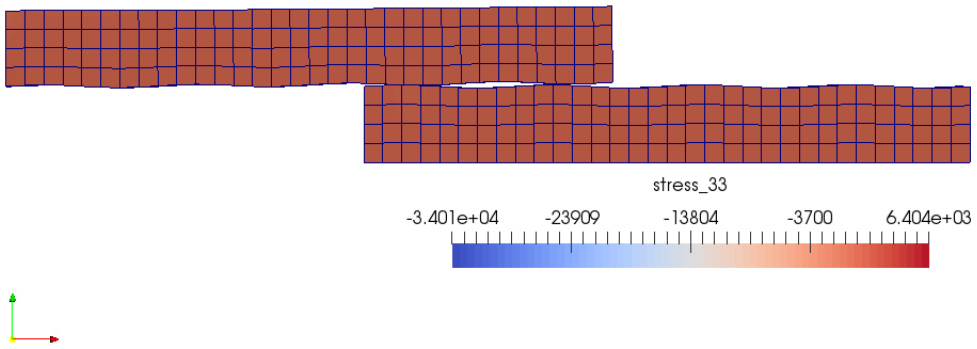

Figure 82: T_{33} plot showing a later stage contact interaction as the left hand side of the bottom brick has slide past the midpoint of the cantilever beam. Note the contact occurs as a crest in the beam's perturbed surface closer to its end and the amplitude of the subsequent vertical displacement is less.

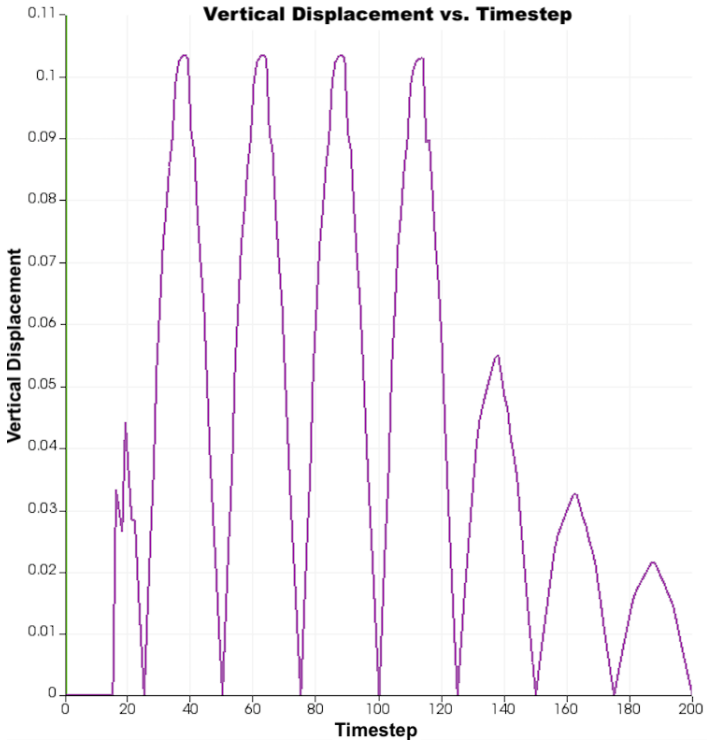

Figure 83: Vertical displacement (meters) vs. timestep plot at the tip of the cantilever.

11.11 Subcycle Performance

This section discusses subcycle performance for two example problems: the stacked cantilever problem presented in Section 11.2.2, which features the unperturbed, conforming meshes at the interface, and the brick compression problem presented in Section 11.7 that features an offset perturbed interface. The first problem is chosen because the tendency for the top beam to "kick" up in localized regions, while other regions along the interface remain in contact, are characteristics that the subcycling method is explicitly designed to handle. The second problem is chosen because the combination of compressive behavior and axial elongation results in a more complex sliding contact interaction along the interface. The end of the simulation, which features the most contact, deformation, and stress, takes more Newton iterations to converge as well as subcycles to resolve the contact interface.

One way in which subcycle performance is assessed is by looking at the number of subcycles per Newton iteration for a given timestep. A subcycle consists of a mode = 5 solve with a mode = 4 convergence check. Recall that there will always be a mode = 2 solution pass with a solve using intermediate $(\alpha_\alpha, \beta_\alpha)$ parameters at each contact element. This is not counted as a subcycle. The number of subcycles per Newton iteration reported then is also the number of mode = 5 solution updates and solves. Another characteristic of the subcycle procedure that we are interested in understanding is the number of times α_α parameters are flipped at each contact overlap for a given Newton iteration. That is, how many times do we flip between contact/no-contact enforcement overall all subcycles for a given Newton iterate? This is reported in two ways. The first is that the average number of flips over all contact elements over all subcycles for each Newton iteration within a single time step is reported. The next is that we report the overlap with the maximum number of flips over all subcycles at each Newton iterate for a single time step. To accompany this data, the percentage of all contact overlaps in the active set that experience an alpha-flip is reported. Additionally, the average number of Newton iterations over *all* timesteps is reported as well

as the average number of subcycles over all Newton iterations over all timesteps. In all, this data helps us to understand how long we are in subcycling per Newton iteration. While percentage of overlaps with alpha-flips is problem and deformation dependent, what we want to observe is low subcycle counts and reasonable Newton iteration counts. This is both out of computational effeciency and cost considerations, but also because this tells us that the subcycling procedure can effectively determine a distribution of contact constraints over all overlaps in the active set that leads to a converged contact and equilibrium solution. This data for the two aforementioned examples is presented in the following sections.

11.11.1 Stacked Cantilever with Displacement Boundary Conditions

The active set for this problem includes all facet-facet overlap pairs along the common interface for all Newton iterations for all timesteps. The difficulty, given the physical behavior of this problem as described in Section 11.2.2, is to determine the exact combination of facet-pairs (i.e. contact elements) that ought to have a zero-gap, postive compressive contact pressure constraints enforced. Even for this problem, where the majority of facet-pairs exhibit zero contact pressure with separation, the subcycling procedure is able to hone in on the exact distribution of facet-pairs over which to enforce contact. Having said that, this section demonstrates that this is done efficiently by examining the subcycle performance at a single timestep.

Here we examine the first out of sixty timesteps in the simulation. The entire contact interface exhibits zero gap with conforming contacting meshes at the beginning of the step; however, the converged end step solution experiences only one contact element actually in contact. This is the end element at the tip where the displacement boundary condition is applied. The other contact elements have very small separation (1.E-6) and zero pressure. Any constraint enforcement method must be able to determine the correct constraint even

in the presence of very small gaps.

The average number of Newton iterations over all timesteps is 1.1 with the first timestep having the most at 5. The average number of subcycles over all Newton iterations is 4.65. Figure 84 shows the number of subcycles per the five Newton iterations for the first timestep. Notice the maximum is 10 at the very first Newton iteration, but as the displacement solution is iterated upon, the subcycles settle to around 3-4 for the rest of the step. Figure 85 shows the average number of alpha-flips (between 0 and 1 for contact/no-contact constraint enforcement) over all subcycles for each Newton iteration over the timestep. Taking the first Newton iteration as an example, even though we have 10 subcycles, which is quite high, we want to see low alpha-flip counts as evidence that any one contact element is not oscillating in-and-out of contact through the subcycling. Obviously an ill-performing constraint enforcement procedure may experience contact elements coming in-and-out of contact, never quite settling on a "converged" pressure/gap solution. Here, we see that the average number of alpha-flips is less than 2 for *all* Newton iterations for this step. Paired with this information is the plot of the maximum number of alpha-flips for any single subcycle for each Newton iteration in the step, which is shown in Figure 86. Note that the maximum number of flips is only 3 for the first Newton iteration, and ≤ 2 for every subsequent Newton iteration. This data demonstrates the efficiency and efficacy of the subcycling procedure. Lastly, Figure 87 shows the percentage of all overlaps in the active set that experience alpha-flips. What we do not want to see are large percentages for all Newton iterations. This would indicate a lack of efficiency in the constraint enforcement method to even initially approximate a correct distribution of constraints across all overlaps. Clearly, starting a mode = 2 solve with an initial intermediate setting of all $(\alpha_\alpha, \beta_\alpha)$ provides an adequate approximation of the interface geometry in the sense that when each α_α parameter is then set to either 0 or 1 in the subsequent mode = 4 solution pass, some of those constraints hold over all subcycles for a given Newton iteration. This is evidenced in Figure 87 in that

all percentages are less than 100. The first Newton iteration, which has the most number of subcycles, has the highest percentage of flips, whereas encouragingly, every subsequent Newton iterations has around 50 percent or less.

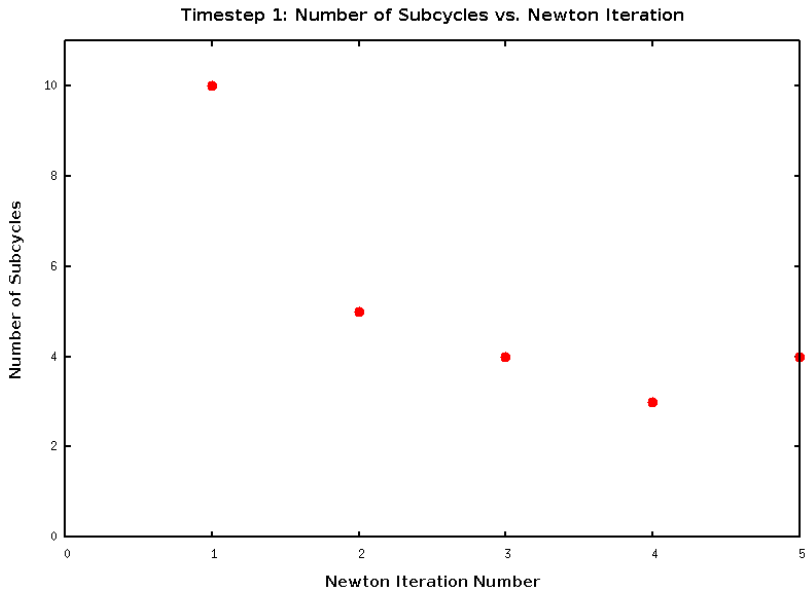

Figure 84: The number of subcycles per Newton iteration for all Newton iterations in the first timestep.

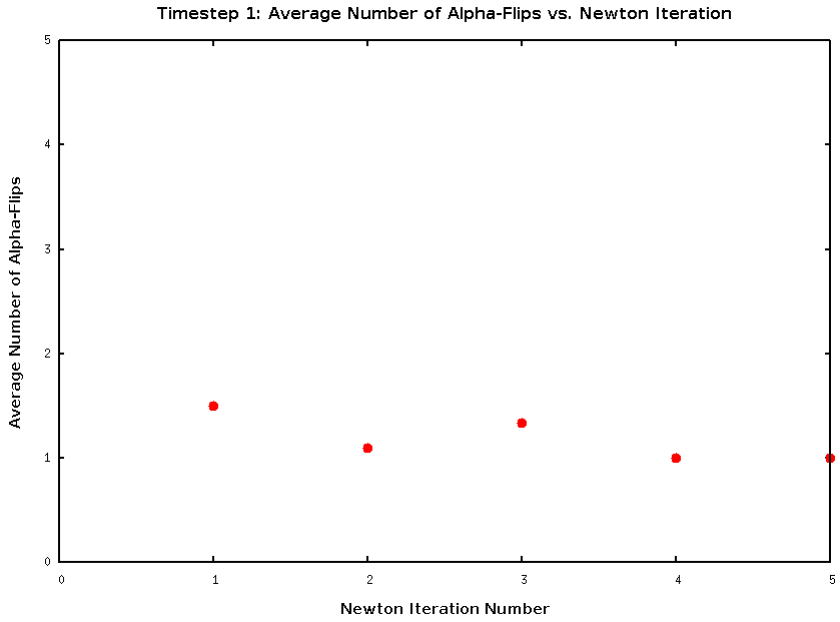

Figure 85: The average number of alpha-flips over all subcycles over all facet-pairs for each Newton iteration in the first timestep.

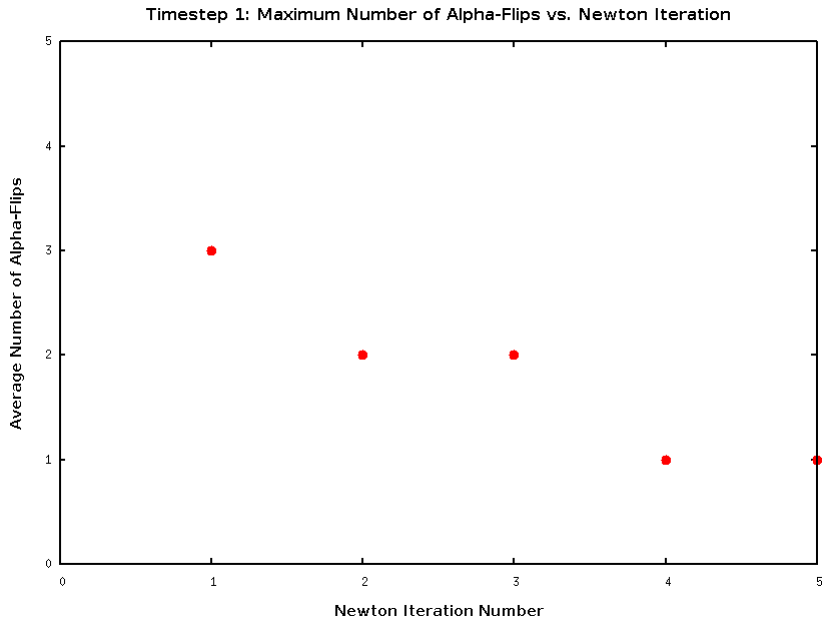

Figure 86: The maximum number of flips out of all subcycles for a given facet-pair at each Newton iteration in the first time step.

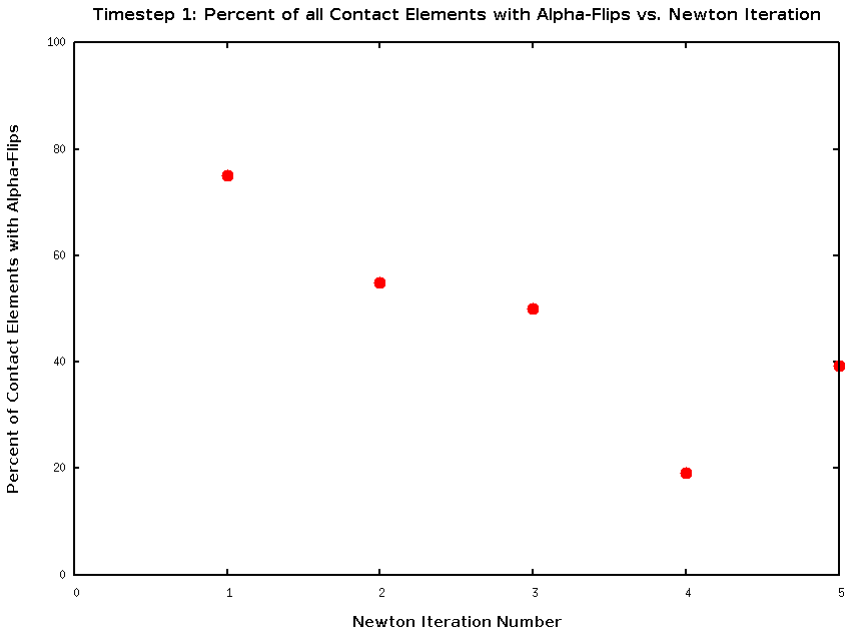

Figure 87: The percentage of all "contacting" facet-pairs that experience alpha-flips during a Newton iteration's subcycling for the first timestep.

11.11.2 Brick Compression

The brick compression problem presented in Section 11.7 consists of one brick coming into contact with, and compressing a second brick. This deformation is complicated by the fact that the contact surfaces are sinusoidally perturbed and offset. The offset results in very localized interaction at the first instance of contact, but as the top brick further displaces downward, the offset forces axial elongation in both bricks and further contact along the interface. The contacting regions along the interface increase due to the sliding induced by the axial elongation until, finally, there is full contact across the interface at the end of the simulation. The reason this problem is difficult is that the two contact surfaces are close enough at the first instance of contact that all facet-pair overlaps are included in the active set. This does not strictly have to be the case, but this was done to fully test the subcycle procedure. That is, all facet-pairs are considered for possible contact, but very few are actually in contact at the beginning of the interaction. As the interaction evolves, the subcycling procedure must determine the additional facet-pairs over which to enforce zero-gap, positive compressive contact pressure, that yields a properly converged contact solution along the interface that also results in equilibrium convergence.

An important note concerns the statement that the entire contact interface is in fact in contact at the end of the simulation. This is, of course, what is meant to be occuring physically as one brick fully compresses into the other brick. Numerically, however, there is a little more to it. Saying that the entire contact interface is in contact does not mean that the entire active set consists of zero-gap, contacting facet-pairs. Factors such as mesh densities paired with the geometry of the interface may preclude this from happening. The job of the subcycle procedure, then, is not just to determine the collection of contacting facet-pairs in a "bloated" active set, but also to determine the exact distribution of contacting facet pairs where geometrically, the entire contact interface is and should strictly be included in the active set. That is, numerical experimentation demonstrated that even if the nodal

velocity projection used to kick off Newton's method at the beginning of a timestep results in all facet-pairs exhibiting interpenetration, the contact solution may not in fact include all facet-pairs in the active set in zero-gap compressive contact. It is as if there is an implicit geometric restriction present simply based on the approximate surface geometry. Any robust constraint enforcement method must handle this. Specifically, let us consider for a moment an initially conforming contact interface. As sliding occurs, as happens in much of the problems presented in this work, one facet on one side of the interface will overlap with two facets on the opposing side of the interface. At the instance of sliding and occurrence of this one-to-two overlap, one of the overlap areas will be much larger than the other overlap area. The subcycle procedure generally favors the larger areas of overlap in contact enforcement, where the smaller area experiences zero contact pressure enforcement. If the subcycling were not to occur, it is very possible to obtain, and in fact there is nothing precluding, a tensile contact solution at these smaller overlaps. As a result, while this brick compression clearly experiences an interface that is fully in contact, only the larger overlaps participate in the compressive contact enforcement. As a result, the subcycle performance for this problem at the last timestep (timestep 100) is examined.

The average number of Newton iterations over all timesteps for this simulation is 1.06 with the last timestep having the most at 4. The average number of subcycles over all Newton iterations for all timesteps is 1.75. To exemplify the aforementioned point about what may occur in a "fully" contacting interface, only about 53 percent of facet-pairs in the active set at the end of the last timestep are in compressive contact with all non-contact facet-pairs exhibiting separation only up to 1.E-5. Even though only about half the active set is in compressive contact, which means the subcycling procedure is in fact working to determine that exact distribution, the number of subcycles per Newton iteration is low. Figure 88 shows that the number of subcycles per Newton iteration is either 4 or 5 with the average number of alpha-flips over all subcycles per Newton iteration held near 1.0 for all Newton

iterations, as shown in Figure 89. Additionally, Figure 90 shows that the maximum number of alpha-flips at any overlap over all subcycles per Newton iteration is 2, which is very low. Furthermore, per Figure 91, the percentage of all overlaps experiencing alpha-flips in order to hone in on a distribution of constraints starts a little above 60 percent and decreases to 50 percent in the last Newton iteration. Recall that the number of contact overlaps actually in contact at the end of the timestep is approximately 50 percent. What this means is that the initial mode = 2 solve, with intermediate settings of $(\alpha_\alpha, \beta_\alpha)$, creates a distribution of contact pressures (albeit some compressive and some tensile) that informs the update of all α_α parameters to either 0 or 1 in the next mode = 4 subcycle update that enforces proper constraints at 50 percent of overlaps. The remaining 50 percent of overlaps, then, are the subject of subcycling updates, of which only 2 maximum alpha-flips are required at any given overlap to arrive at a converged distribution of constraints that not only provides an acceptable contact solution, but yields equilibrium convergence.

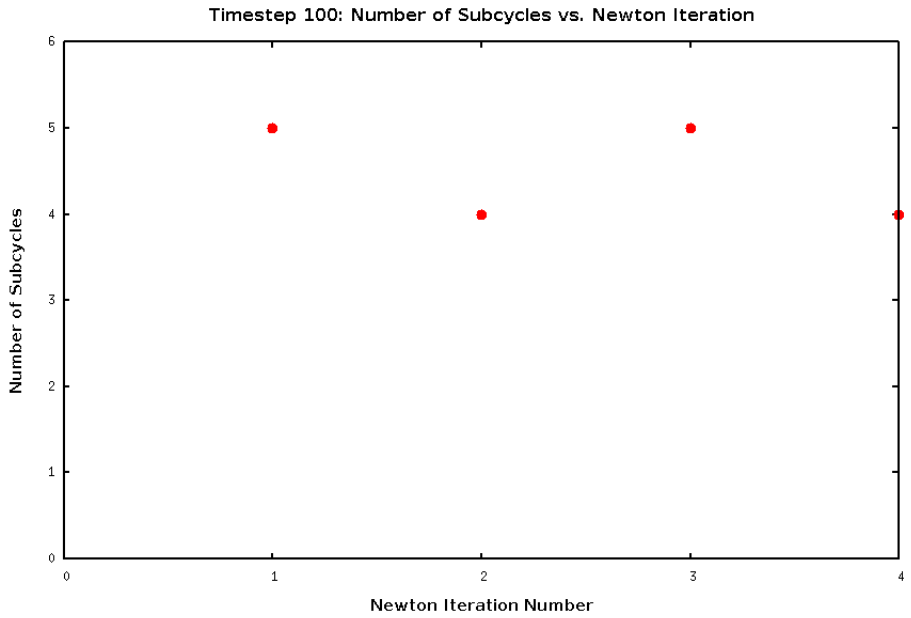

Figure 88: The number of subcycles per Newton iteration for all Newton iterations in the last timestep.

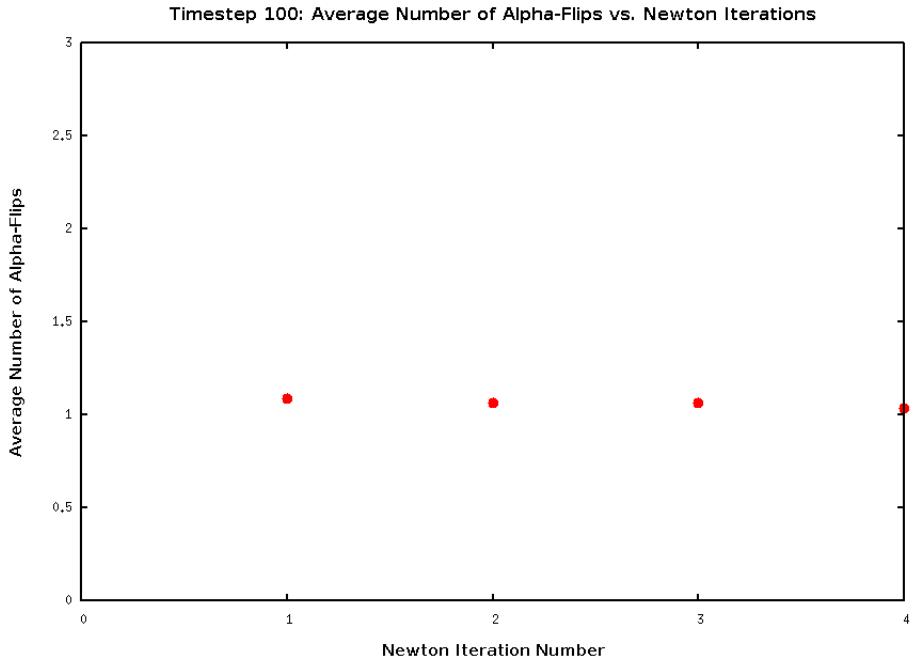

Figure 89: The average number of alpha-flips over all subcycles over all facet-pairs for each Newton iteration in the last timestep.

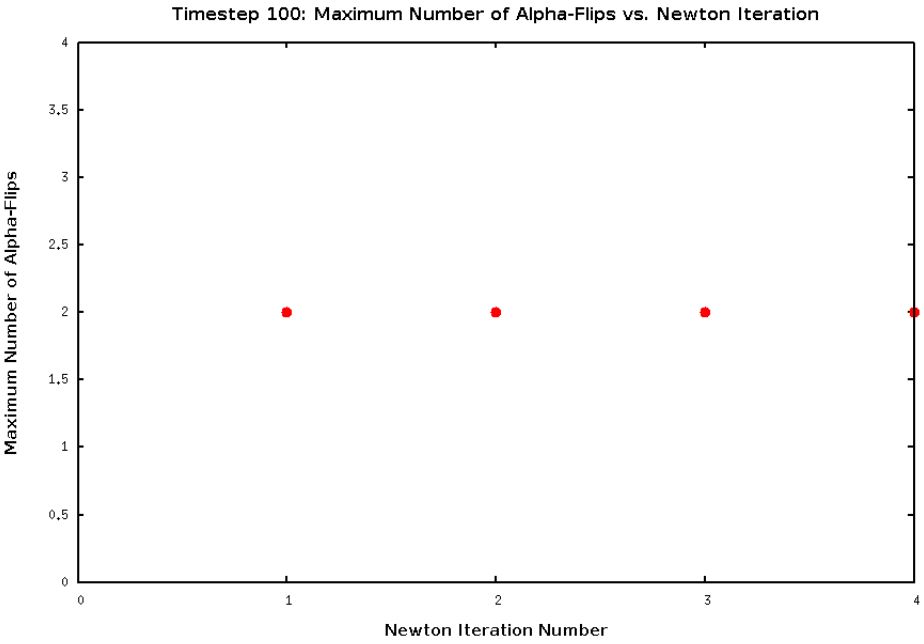

Figure 90: The maximum number of flips out of all subcycles for a given facet-pair at each Newton iteration in the last time step.

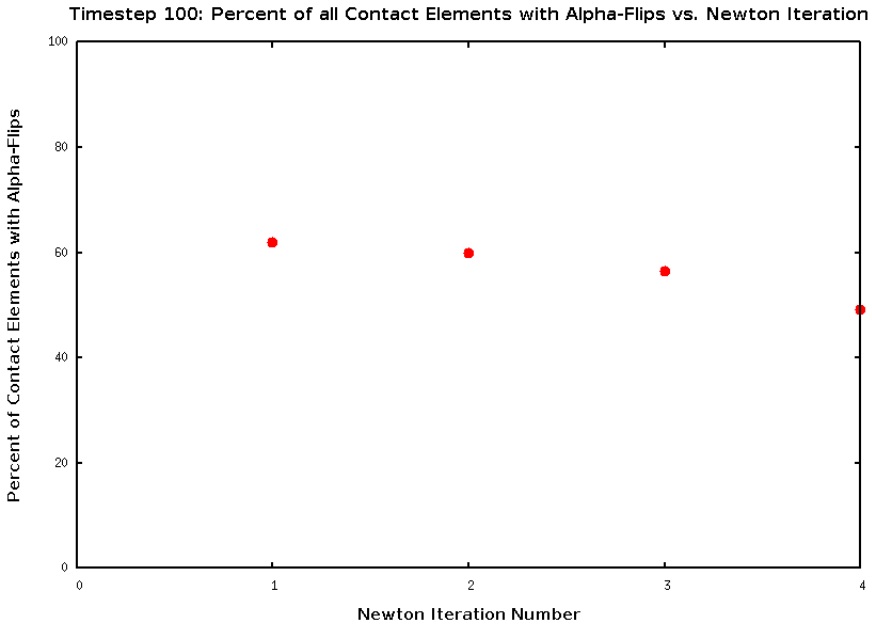

Figure 91: The percentage of all "contacting" facet-pairs that experience alpha-flips during a Newton iteration's subcycling for the last timestep.

12 Future Work

First and foremost the work presented herein does not include a frictional contact implementation. A discussion is provided in Section 10.6 including the mathematical framework placing a friction implementation squarely within the structure of the subcycling method presented in Section 10.4. Nevertheless, the implementation and numerical experimentation supporting the frictional framework is required.

Additionally, study of the pressure basis is an open area of research. While a constant pressure basis is often adequate to enforce contact, and does in fact preclude any instabilities in the solution, the effects of a linear pressure basis ought to be studied. A linear pressure basis may be particularly advantageous when considering contacting surfaces with irregular and distorted meshes with a high degree of non-conformity between the two. Numerical experimentation will be required to see how the construction of the pressure basis (used to preclude solution instabilities) interacts with the subcycle procedure. It is possible that some

algorithmic modifications may be required as new details of the implementation come to light with a higher order pressure basis. The purpose of this study would be to understand if the construction of a pressure basis using discrete linear pressure polynomials per Section 9.3 increases the flexibility and efficacy of the method in solving difficult contact problems.

A final area for further development is to extend the tope-on-tope interactions currently available in the implementation presented in this work. This contact methodology was implemented in 3D using eight node hexahedral elements, which involve four node quadrilateral faces. The data structures were designed to flexibly handle other tope-on-tope interactions, such as three node triangle to four node quadrilateral, or even single node to four node quadrilateral face enforcement. Of course, a node-to-face enforcement is a departure from the face-on-face methodology, but there are geometric limitations where a particular discrete interaction may be resolved more effectively by a node-to-face interaction or even a segment-to-face interaction where a 1D line segment interacts with a 2D face. One can imagine a brick sliding off a cliff, for example. As the brick teeters, prior to falling, the contact interaction is explicitly a segment-to-face interaction. These ideas represent full geometric flexiblity in handling contact enforcement. In the case of a node-to-face or segment-to-face interaction, which would be enforced on a geometrically specific basis, an alternative traction/force formulation will be required. Additional geometric algorithms will also be required to adequately assess and enforce the correct contact methodology per tope-on-tope interaction.

13 Conclusions

This work has presented a novel face-on-face contact formulation for nonlinear solid mechanics using finite elements in 3-dimensions. The formulation and implementation uses four node quadrilateral faces belonging to eight node hexahedral volume elements, but is

sufficiently general and can be applied to any face-on-face interaction using linear finite elements. The contact interface is discretized using a symmetric median-plane methodology and the contact constraint enforcement uses an integral gap expression and is performed using a novel subcycling procedure. This procedure is one of the main contributions in this work. The procedure treats active and non-active constraints simultaneously, allowing for more robust treatment of interface geometry. This procedure also uses multiple rank-one updates to factor the system of equations as a proof-of-concept implementation of an effective solver procedure. Furthermore, while the numerical work presented uses a constant contact pressure basis, the theoretical framework and algorithmic implementation are designed to handle up to a linear pressure basis. Studying the effects of a linear pressure basis, as well as any inf-sup-type instabilities, is fertile ground for future research.

The numerical implementation of the proposed contact method was performed in an implicit nonlinear solid mechanics code. While the numerical examples presented in this work are quasi-static contact problems, the contact formulation itself is general. As such, contact using implicit dynamics is a straightforward modification to the current implementation. The numerical examples demonstrate the efficacy of the proposed method in the presence of material and geometric nonlinearity, as well as large sliding. Moreover, the examples show the successful performance of the subcycling constraint enforcement procedure. This procedure handles complex interface geometries through the proper selection of active and non-active constraints, leading to converged contact solutions. This procedure places no restrictions on the formation of the set of contact face-pairs, does not rely on heuristics in selecting active constraints, and precludes nonphysical solutions even in the presence of over-populated sets of face-pairs. Thus, significant flexiblity is introduced, allowing for the solution of difficult quasi-static contact problems.

References

[1] M. Rashid, "Eci 201: Introduction to the theory of elasticity," *UC Davis Course Reader*, 2012.

[2] M. M. Rashid, "Incremental kinematics for finite element applications," *International Journal for Numerical Methods in Engineering*, vol. 36, no. April, pp. 3937–3956, 1993.

[3] O. Hafez, "Implementation of the compressible mooney-rivlin material in the imitor finite element code." 2016.

[4] H. M. Hilber, T. J. Hughes, and R. L. Taylor, "Algorithms in structural dynamics," *Earthquake Engineering and Structural Dynamics*, vol. 5, no. June 1976, pp. 283–292, 1977.

[5] N. Newmark, "A method of computation for structural dynamics," *Journal of the Engineering Mechanics Division, ASCE*, pp. 67–94, 1959.

[6] T. Laursen, *Computational Contact and Impact Mechanics*. New York: Springer Berlin Heidelberg, 2002.

[7] J. Hallquist, G. Goudreau, and D. Benson, "Sliding interfaces with contact-impact in large-scale lagrangian computations," *Computer methods in applied mechanics and engineering*, vol. 51, pp. 107–137, 1985.

[8] D. Benson and J. Hallquist, "A single surface contact algorithm for the post-buckling analysis of shell structures," *Computer methods in applied mechanics and engineering*, vol. 78, pp. 141–163, 1990.

[9] 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*, vol. 193, pp. 601–629, 2003.

[10] P. Wriggers, *Computational Contact Mechanics Second Edition*. Springer Berlin Heidelberg, 2006.

[11] P. Wriggers, L. Krstulovic-Opara, and J. Korelc, "Smooth c1-interpolations for two-dimensional frictional contact problems," *International Journal for Numerical Methods in Engineering*, vol. 51, pp. 1469–1495, 2001.

[12] T. Belytschko, W. Daniel, and G. Ventura, "A monolithic smoothing-gap algorithm for contact-impact based on the signed distance function," *International Journal for Numerical Methods in Engineering*, vol. 55, pp. 101–125, 2002.

[13] M. A. Puso and T. A. Laursen, "A 3d contact smoothing method using gregory patches," *International Journal for Numerical Methods in Engineering*, vol. 54, pp. 1161–1194, 2002.

[14] J. Gregory, *Computer Aided Geometric Design*, ch. Smooth interpolation without twist constraints, pp. 71–87. New York: Academic Press Inc., 1983.

[15] J. Oliver, S. Hartmann, J. Cante, R. Weyler, and J. Hernandez, "A contact domain method for large deformation frictional contact problems. part 1: Theoretical basis," *Computer methods in applied mechanics and engineering*, vol. 198, pp. 2591–2606, 2009.

[16] S. Hartmann, J. Oliver, R. Weyler, J. Cante, and J. Hernandez, "A contact domain method for large deformation frictional contact problems. part 2: Numerical aspects," *Computer methods in applied mechanics and engineering*, vol. 198, pp. 2607–2631, 2009.

[17] C. Bernardi, Y. Maday, and A. Patera, *Asymptotic and Numerical Methods for Partial Differential Equations with Critical Parameters*, ch. Domain decomposition by the mortar element method. Dordrecth, Netherlands: Kluwer Academic Publisher, 1993.

[18] T. McDevitt and T. Laursen, "A mortar-finite element formulation for frictional con-

tact problems," *International Journal for Numerical Methods in Engineering*, vol. 48, pp. 1525–1547, 2000.

[19] M. A. Puso and T. A. Laursen, "A mortar segment-to-segment frictional contact method for large deformations," *Computer methods in applied mechanics and engineering*, vol. 193, pp. 4891–4913, 2004.

[20] M. A. Puso, T. Laursen, and J. Solberg, "A segment-to-segment mortar contact method for quadratic elements and large deformations," *Computer methods in applied mechanics and engineering*, vol. 197, pp. 555–566, 1008.

[21] A. Popp, M. W. Gee, and W. A. Wall, "A finite deformation mortar contact formulation using a primal-dual active set strategy," *International Journal for Numerical Methods in Engineering*, vol. 79, pp. 1354–1391, 2009.

[22] A. Popp, M. Gitterle, M. W. Gee, and W. A. Wall, "A dual mortar approach for 3d finite deformation contact with consistent linearization," *International Journal for Numerical Methods in Engineering*, vol. 83, pp. 1428–1465, 2010.

[23] M. Tur, F. Fuenmayor, and P. Wriggers, "A mortar-based frictional contact formulation for large deformations using lagrange multipliers," *Computer methods in applied mechanics and engineering*, vol. 198, pp. 2860–2873, 2009.

[24] G. Strang, *Introduction to Linear Algebra*. Wellesley-Cambridge Press, 5th ed., 2016.

[25] M. Rashid, "The arbitrary local mesh replacement method: An alternative to remeshing for crack propagation analysis," *Computer methods in applied mechanics and engineering*, vol. 154, pp. 133–150, 1998.

[26] M. A. Taylor, B. A. Wingate, and L. P. Bos, "Several new quadrature formulas for polynomial integration in the triangle," *arXiv:math/0501496v2 [math.NA]*, p. 14, 2007.

[27] M. Metcalf, J. Reid, and M. Cohen, *Moder Fortran explained*. Oxford University Press, 2011.

[28] V. A. Yastrebov, *Numerical Methods In Contact Mechanics*. John Wiley and Sons, Inc., 2013.

[29] W. Fujun, C. Jiangang, and Y. Zhenhan, "A contact searching algorithm for contact-impactor problems," *Acta Mechanica Sinica (English series)*, vol. 16, pp. 374–382, 2000.

[30] J. Williams and R. O'Conner, "Discrete element simulation and the contact problem," *Archives of Computational Methods in Engineering*, vol. 6, pp. 279–304, 1999.

[31] B. Yang and T. Laursen, "A contact searching algorithm including bounding volume trees applied to finite sliding mortar formulations," *Computational Mechanics*, vol. 41, pp. 189–205, 2008.

[32] B. Yang and T. Laursen, "A large deformation mortar formulation of self contact with finite sliding," *Computer Methods in Applied Mechanics and Engineering*, pp. 756–772, 2008.

[33] P. Gill, G. Golub, W. Murray, and M. Saunders, "Methods for modifying matrix factorizations," *Math. Comp.*, vol. 28, no. 126, pp. 505–535, 1974.

[34] L. Deng, *Multiple-Rank Updates to Matrix Factorizations for Nonlinear Analysis and Circuit Design*. PhD thesis, Stanford, 2010.

[35] J. Simo and T. Hughes, *Computational Inelasticity*. Springer, 1998.

[36] C. Truesdell and W. Noll, *The Non-Linear Field Theories of Mechanics*. Springer-Verlag Berlin Heidelberg, 3 ed., 2004.

[37] M. Yip, Z. Li, B.-S. Liao, and J. Bolander, "Irregular lattice models of fracture of multiphase particulate material," *International Journal of Fracture*, vol. 140, pp. 113–124, 2006.

[38] A. W. Naylor and G. R. Sell, *Linear Operator Theory in Engineering and Science*. Springer-Verlag, 1982.

[39] H. F. Davis and A. D. Snider, *Introduction to Vector Analysis*. Hawkes Publishing, seventh ed., 2000.

[40] J. Fish and T. Belytschko, *A First Course in Finite Elements*. John Wiley and Sons, Inc., 2012.

[41] M. Stadler and G. Holzapfel, "Subdivision schemes for smooth contact surfaces of arbitrary mesh topology in 3d," *International Journal for Numerical Methods in Engineering*, vol. 60, no. 7, pp. 1161–1195, 2004.

[42] P. Wriggers and G. Zavarise, "A formulation for frictionless contact problems using a weak form introduced by Nitsche," *Computational Mechanics*, vol. 41, pp. 407–420, July 2007.

[43] M. M. Rashid and A. Sadri, "The partitioned element method in computational solid mechanics," *Computer Methods in Applied Mechanics and Engineering*, vol. 240, pp. 152–165, 2012.

[44] G. Kloosterman, R. M. J. van Damme, A. H. van den Boogaard, and J. Huétink, "A geometrical-based contact algorithm using a barrier method," *International Journal for Numerical Methods in Engineering*, no. February 2000, pp. 865–882, 2001.

[45] T. J. Hughes, *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*. Mineola, New York: Dover Publications, Inc., 2000.

[46] P. Chadwick, *Continuum Mechanics: Concise Theory and Problems*. London: George Allen & Unwin Ltd., 1999.

[47] C. Bernardi, N. Debit, and Y. Maday, "Coupling finite elements and spectral methods: first results," *Math. Comput.*, vol. 54, pp. 21–39, 1990.

[48] R. Asaro and V. Lubarda, *Mechanics of Solids and Materials*. New York: Cambridge University Press, 2006.

[49] M. W. Mahoney and P. Drineas, "Cur matrix decompositions for improved data analysis," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 106, no. 3, pp. 697–702, 2012.

[50] T. A. Laursen and J. Simo, "An augmented lagrangian treatment of contact problems involving friction," *Computers and Structures*, vol. 42, pp. 97–116, 1992.

[51] R. A. Brualdi, S. Friedland, and A. Pothen, "The sparse basis problem and multilinear algebra," *Society for Industrial and Applied Mathematics*, vol. 16, pp. 1–20, 1993.

[52] B. Recht, M. Fazel, and P. A. Parrilo, "Guaranteed minimum-rank solutions for linear matrix equations via nuclear norm minimization," *Society for Industrial and Applied Mathematics*, vol. 52, pp. 471–501, 2010.

[53] V. Ozoliņš, R. Lai, R. Caflisch, and S. Osher, "Compressed modes for variational problems in mathematics and physics," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 110, no. 46, pp. 18368–18373, 2014.