SAND2019-12930

SANDIA REPORT

Printed

Sandia National Laboratories

An error estimation driven adaptive tetrahedral workflow for full engineering models

Brian N. Granzow, Alejandro Mota, Daniel Alejandro Ibanez, James W. Foulk III

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185
Livermore, California 94550

Issued by Sandia National Laboratories, operated for the United States Department of Energy by National Technology & Engineering Solutions of Sandia, LLC.

Available to DOE and DOE contractors from

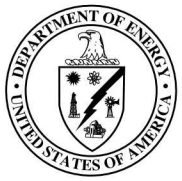
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865) 576-8401
Facsimile: (865) 576-5728
E-Mail: reports@osti.gov
Online ordering: http://www.osti.gov/scitech

Available to the public from

U.S. Department of Commerce
National Technical Information Service
5301 Shawnee Road
Alexandria, VA 22312

Telephone: (800) 553-6847
Facsimile: (703) 605-6900
E-Mail: orders@ntis.gov
Online order: https://classic.ntis.gov/help/order-methods



ABSTRACT

Tetrahedral finite element workflows have the potential to drastically reduce time to solution for computational solid mechanics simulations when compared to traditional hexahedral finite element analogues. A recently developed, higher-order composite tetrahedral element has shown promise in the space of incompressible computational plasticity. Mesh adaptivity has the potential to increase solution accuracy and increase solution robustness. In this work, we demonstrate an initial strategy to perform conformal mesh adaptivity for this higher-order composite tetrahedral element using well-established mesh modification operations for linear tetrahedra. We propose potential extensions to improve this initial strategy in terms of robustness and accuracy.

ACKNOWLEDGMENT

We are grateful for the opportunity to participate in the Director's Challenge. The challenge does resonate with our team and we look forward to future opportunities. We appreciate the interest of the Engineering Science IAT and thank Arthur Brown for his generous support. Glen Hansen supported our work through the entirety of the process and has been instrumental in helping our team be successful. It is our hope that our contributions are commensurate with his enthusiasm. We would also like to thank staff members Jake Ostien and Brandon Talamini. Jake was immensely supportive and his initial advice regarding implementation was extremely helpful. Brandon helped the team think about fundamental tradeoffs in implementing the internal force in the reference or current configuration. With limited funding, one can only be successful when both managers and staff members come together to support a vision that is appealing to our community. We retain that vision.

Supported by the Laboratory Directed Research and Development program at Sandia National Laboratories, a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA-0003525.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. MOTIVATION

Significant strides have been recently made in the development, implementation, rapid discretization, and remeshing/mapping of higher-order tetrahedral elements suitable for resolving large, plastic deformations in solid mechanics. New tetrahedral workflows have hastened CAD-to-Solution process for geometrically complex components by 100X. Our current workflow, however, will be prohibitively expensive for system-level models to sufficiently predict engineering responses which span both physics and length/time scales.

Although new efforts in Next Generation Simulation (NGS) are addressing parallel discretization, little effort has been devoted to error estimation and local adaptivity. We assert that foundational efforts in error estimation/indication for transient, inelastic processes will provide the technical basis for in-core mesh adaption in full engineering models necessary for the design, development, and qualification of systems.

A posteriori error estimation provides the critical feedback needed for mesh adaption and the fulfillment of Steve Younger's vision for systematic modeling of relevant phenomena.

In an effort to illustrate both the need and the utility of error estimators/indicators in abnormal mechanical environments, we have opted to develop a posteriori error estimation in a new shock multiphyiscs application termed Lagrangian Grid Reconnection LGR. In addition to being lightweight, LGR is built upon Omega_h, a portably performant library for adapting triangular and tetrahedral meshes using anisotropic metrics. Our goal is to develop and illustrate methods for error estimation/indication that drive adaptivity for transient, evolutionary processes that encompass the localization of the deformation. The demonstration of in-core workflow capable of resolving large inelastic deformations via mesh adaptivity and the mapping of internal variables for dynamic, inelastic processes provides a critical, first step towards the affordable resolution of full engineering models.

To leverage recent work on the composite tetrahedral element, we plan to implement and verify the five-field composite-tet formulation intended for isochoric deformations into LGR. In addition, recent advances in mapping internal variables via Lie Algebras will also be verified. Team members in 8363 and 1540 seek to partner with team members in 1443 to ensure commonality with the Sierra/SM production environment in constitutive models, element technology, and mapping methodologies.

Because error estimation without adaptivity is akin to words without action, we believe that incorporating our best technologies in error estimation, element technology, and mapping into current products featuring in-core adaptivity is a viable path forward. The totality of the infrastructure illustrating the refinement and coarsening inherent in capturing localized processes will realize the potential of a new, adaptive tetrahedral workflow capable of

systematically modeling dominant phenomena. In prior work, we focused on the burden of discretization. Affordable, system-level approaches require that we achieve solutions with minimal computational cost to permit trade-off studies for design and the necessary V&V for qualification. Hence, the team seeks to maintain the resolution of dynamic, localized processes with a 10X reduction in the degrees of freedom for solution. The new, affordable tetrahedral workflow is a critical step in realizing our Director's vision.

2. APPROACH

Recent success in the application of a new tetrahedral workflow for solid mechanics in our production environment for NW components have confirmed the potential of a tetrahedral workflow to ease the burden of discretization. Component analysts are able to achieve solutions in days, not months. Although we continue to make fundamental improvements in element technology, contact algorithms, and discretization, we believe that our current foundation of Sierra/SM and Cubit can provide value to many ND analysts today. Interactions with system-level analysts, however, have revealed a multitude of needs that primarily focus on model size. They simply cannot afford isotropic discretizations with a constant element size. Although incorporating varying size fields and anisotropy into current discretization processes is prudent and is currently being pursued, we hypothesize that the research opportunity is to incorporate error estimators/indicators that drive anisotropic adaptivity. If we can maintain solution quality with a 10X reduction in the degrees of freedom for solution in the first year of our project, we believe that our future work can establish the technical basis to realize the Director's vision.

To be clear, the call seeks for researchers to "demonstrate an innovative method to solve coupled PDEs on complex 3D geometries without having to manually create a mesh." We believe that our team can already accomplish this task today in our production environment. Analysts in 1556 are rapidly employing a tetrahedral workflow on complex 3D geometries to provide solutions to NW customers in FY18. From our perspective, widespread adoption to full engineering models requires that we innovate to automate affordable predictions.

To realize the vision of the ES IAT with limited funds on a compacted time line, we have elected to leverage and develop with established methods and tools

1. Employ the current 5-field composite tetrahedral element suitable for elastoplasticity
2. Leverage mapping technology for internal variables
3. Develop within the lightweight shock multiphysics application LGR built on Omega_h
4. Leverage Omega_h, an open source C++ library for mesh adaption

and research new avenues with contributions in

1. Developing new error estimators in explicit dynamics for elastoplastic solids under large deformations. We will provide the necessary anisotropic metrics for Omega_h to drive adaptivity.
2. Incorporating the aforementioned developments in element technology and mapping into LGR.

We believe this approach to be the most direct path to realize our research opportunity and form the multidisciplinary team requisite for understanding. New error estimators. A posteriori error estimation routines can be categorized as follows, in order of least complex to most complex: recovery-based, residual-based, and adjoint-based. Recovery-based and residual-based error estimation techniques attempt to bound or approximate the discretization error in a chosen global norm, while adjoint-based procedures approximate the error in a functional quantity of interest.

The history and development of each of these techniques is intimately linked with solid mechanics applications. Despite this rich history, though, a posteriori error estimation has remained relatively unexplored for (a) dynamic (b) nonlinear solid mechanics applications with (c) history-dependent variables. Works that investigate applications with all three properties include estimating the discretization error in a global norm based on a principle of minimum potential energy [38, 29] and using geometric multigrid techniques [4].

It is, however, more common to find methods developed for some subset of the properties (a)-(c). For instance, in the context of quasi-static mechanics applications with history dependent variables, recovery-based error procedures have been used in metal forming simulations [5], residual-based error estimation techniques have been used to study viscoplasticity [9] and nonlocal damage models [41], and adjoint-based error estimates have been applied to studies employing perfect plasticity [39]. In the context of truly transient phenomena lacking history dependent variables, the so-called constitutive-relation error [21, 7], recovery-based approaches [22], and adjoint-based techniques [45] have been used to study linear elastodynamics. Adjoint-based techniques have also been utilized to estimate functional errors in quasi-static nonlinear elasticity applications [23, 46, 14]. Lastly, there has been recent interest in using the variational multiscale (VMS) method, which is naturally equipped with a residual-based point wise error estimate [15], to control discretization errors in quasi-static finite elasticity [28].

Each of the error estimation techniques utilized in the previous literature show promise in the space of explicit transient dynamics for elastoplastic solids. However, because this space is relatively unexplored in the context of error estimation and mesh adaptation, the practicality and utility of each approach is as of yet unknown. To map the space of applicability for each of these types, we propose implementing a new library for a posteriori error estimation in that will combine error information with directional solution information to provide Omega_h with anisotropic metric inputs to drive mesh adaptation. We emphasize that we will leverage anisotropic adaptation, a technique not commonly employed in solid mechanics, to achieve the goal of a 10X reduction in degrees of freedom while still maintaining solution fidelity.

Initial developments will include implementing practical recovery-based procedures, such as Zienkiewicz-Zhu and Hessian-based error approximation techniques to quickly demonstrate the ability to perform in-core adaptive workflows for explicit transient dynamics. Rather that focus on oscillatory state variables such as the Cauchy stress, we will gravitate towards monotonically increasing internal state variables. In conjunction, more rigorous residual-based error estimates and VMS-based error estimates will be mathematically studied and then implemented into the error estimation library. With the ability to switch

between different types of error estimates, the space of applicability of each type of estimate will be explored for the use cases aligned with abnormal mechanical environments.

Beyond FY19, we enter the iterative process of assessing and improving the reliability and robustness of both error estimation and mesh adaptation for numerous use cases with a focus on the localization of the deformation. Longer term efforts include the practicality and ability to perform adjoint-based error estimation. The in-core workflows realized through Omega_h, LGR, and the newly developed error estimation library will allow us to explore extending adjoint-based techniques from quasi-statics to dynamics with history dependent variables coupled to fully anisotropic mesh adaptation. Such an extension is an immense challenge for the targeted applications, and its feasibility is, as of present, an open research topic.

3. FORMULATION OF CONTINUUM SOLID DYNAMICS

Consider a body as the open set $\Omega \subset \mathbb{R}^3$ undergoing a motion described by the mapping $\boldsymbol{x} = \boldsymbol{\varphi}(\boldsymbol{X}, t) : \Omega \times [0, T] \to \mathbb{R}^3$, where $\boldsymbol{X} \in \Omega$, $t \in [0, T]$, $T \in \mathbb{R}_{>0}$, and $[0, T]$ is the time interval. Assume that the boundary of the body is $\partial\Omega = \partial_{\boldsymbol{\varphi}}\Omega \cup \partial_{\boldsymbol{T}}\Omega$ with unit normal $\boldsymbol{N}$, where $\partial_{\boldsymbol{\varphi}}\Omega$ is a displacement boundary, $\partial_{\boldsymbol{T}}\Omega$ is a traction boundary, and $\partial_{\boldsymbol{\varphi}}\Omega \cap \partial_{\boldsymbol{T}}\Omega = \emptyset$. The prescribed boundary displacements or Dirichlet boundary conditions are $\boldsymbol{\chi} : \partial_{\boldsymbol{\varphi}}\Omega \times [0, T] \to \mathbb{R}^3$. The prescribed boundary tractions or Neumann boundary conditions are $\boldsymbol{T} : \partial_{\boldsymbol{T}}\Omega \times [0, T] \to \mathbb{R}^3$. Let $\boldsymbol{F} := \operatorname{Grad} \boldsymbol{\varphi}$ be the deformation gradient. Let also $R\boldsymbol{B} : \Omega \to \mathbb{R}^3$ be the body force, with R the mass density in the reference configuration. Furthermore, introduce the kinetic energy of the body as

$$T(\dot{\boldsymbol{\varphi}}) := \frac{1}{2} \int_\Omega R\dot{\boldsymbol{\varphi}} \cdot \dot{\boldsymbol{\varphi}} \ \mathrm{d}V, \tag{3.1}$$

and its potential energy as

$$V(\boldsymbol{\varphi}) := \int_\Omega A(\boldsymbol{F}, \boldsymbol{Z}) \ \mathrm{d}V - \int_\Omega R\boldsymbol{B} \cdot \boldsymbol{\varphi} \ \mathrm{d}V - \int_{\partial_{\boldsymbol{T}}\Omega} \boldsymbol{T} \cdot \boldsymbol{\varphi} \ \mathrm{d}S, \tag{3.2}$$

in which $A(\boldsymbol{F}, \boldsymbol{Z})$ is the Helmholtz free-energy density and $\boldsymbol{Z}$ is a collection of internal variables. The Lagrangian function of the body is then

$$L(\boldsymbol{\varphi}, \dot{\boldsymbol{\varphi}}) := T(\dot{\boldsymbol{\varphi}}) - V(\boldsymbol{\varphi}), \tag{3.3}$$

which gives rise to the action functional

$$I[\boldsymbol{\varphi}] := \int_0^T L(\boldsymbol{\varphi}, \dot{\boldsymbol{\varphi}}) \ \mathrm{d}t. \tag{3.4}$$

According to the Variational Principle of Hamilton, the equation of motion is obtained by finding the critical point of the action functional $I[\boldsymbol{\varphi}]$ over the Sobolev space $W_2^1(\Omega)$ that is comprised of all functions that are square-integrable and have square-integrable first derivatives [27]. Define

$$\mathcal{S} := \left\{ \boldsymbol{\varphi} \in W_2^1(\Omega) : \boldsymbol{\varphi} = \boldsymbol{\chi} \text{ on } \partial_{\boldsymbol{\varphi}}\Omega \times [0, T] \right\} \tag{3.5}$$

and

$$\mathcal{V} := \left\{ \boldsymbol{\xi} \in W_2^1(\Omega) : \boldsymbol{\xi} = \boldsymbol{0} \text{ on } \partial_{\boldsymbol{\varphi}}\Omega \times [0, T] \right\} \tag{3.6}$$

where $\boldsymbol{\xi} \in \mathcal{V}$ is a test function. This leads to

$$\begin{aligned}
\delta I := DI[\boldsymbol{\varphi}](\boldsymbol{\xi}) &= \int_0^T \left(\frac{\partial L}{\partial \boldsymbol{\varphi}} \cdot \boldsymbol{\xi} + \frac{\partial L}{\partial \dot{\boldsymbol{\varphi}}} \cdot \dot{\boldsymbol{\xi}} \right) \, \mathrm{d}t = \int_0^T \left(\frac{\partial L}{\partial \boldsymbol{\varphi}} - \frac{\mathrm{d}}{\mathrm{d}t} \frac{\partial L}{\partial \dot{\boldsymbol{\varphi}}} \right) \cdot \boldsymbol{\xi} \, \mathrm{d}t \\
&= \int_0^T \left[\int_\Omega (R\boldsymbol{B} \cdot \boldsymbol{\xi} - \boldsymbol{P} : \operatorname{Grad} \boldsymbol{\xi}) \, \mathrm{d}V + \int_\Omega R\ddot{\boldsymbol{\varphi}} \cdot \dot{\boldsymbol{\xi}} \, \mathrm{d}V + \int_{\partial_{\boldsymbol{T}}\Omega} \boldsymbol{T} \cdot \boldsymbol{\xi} \, \mathrm{d}S \right] \mathrm{d}t \\
&= \int_0^T \left[\int_\Omega (\operatorname{Div} \boldsymbol{P} + R\boldsymbol{B} - R\ddot{\boldsymbol{\varphi}}) \cdot \boldsymbol{\xi} \, \mathrm{d}V + \int_{\partial_{\boldsymbol{T}}\Omega} \boldsymbol{T} \cdot \boldsymbol{\xi} \, \mathrm{d}S \right] \mathrm{d}t = 0,
\end{aligned} \tag{3.7}$$

where $\boldsymbol{P} = \partial A / \partial \boldsymbol{F}$ denotes the first Piola-Kirchhoff stress. The Euler-Lagrange equation correspondng to 3.4 is then

$$\begin{aligned}
\operatorname{Div} \boldsymbol{P} + R\boldsymbol{B} &= R\ddot{\boldsymbol{\varphi}}, \quad \text{in} \quad \Omega \times [0,T], \\
\boldsymbol{P}\boldsymbol{N} &= \boldsymbol{T}, \quad \text{on} \quad \partial_{\boldsymbol{T}}\Omega \times [0,T], \\
\boldsymbol{\varphi} &= \boldsymbol{\chi}, \quad \text{on} \quad \partial_{\boldsymbol{\varphi}}\Omega \times [0,T].
\end{aligned} \tag{3.8}$$

We solve the governing equations with a single-field (displacement-based) Galerkin finite element method using an explicit central difference time integration scheme. This finite element scheme is exactly the same as used in SIERRA/SM [43]. The details are omitted for the sake of brevity. Section 6.2 highlights practically exact agreement between SIERRA/SM and our implementation in LGR.

4. A 10-NODE COMPOSITE TETRAHEDRON

In this work, we concern ourselves entirely with a 10-noded composite tetrahedral finite element [11]. At a high-level, this element is based on a composition of twelve sub-tetrahedra with a piecewise linear basis defined over the sub-tetrahedra and an assumed linear gradient space over the entire composite tetrahedron. The element was developed in the context of finite deformation plasticity, where incompressibility often greatly hinders other tetrahedral finite elements, by considering a Hu-Washizu variational principle.

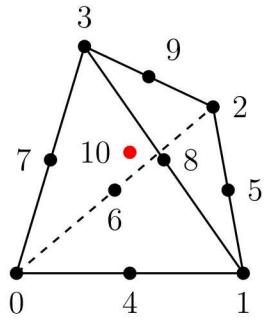


Figure 4-1. The 10-node composite tetrahedral element

Figure 4-1 illustrates the node ordering for nodes 0 through 9 for the 10-node composite tetrahedral element, where the 11th node, labeled 10, is an auxiliary node that is implicitly defined in the element formulation and does not carry with it any degrees of freedom. This node is defined as the average location of the six edge nodes. Table 4-1 demonstrates the connectivity of the twelve sub-tetrahedra in terms of the nodes that define them.

Sub-tetrahedron	Nodes			
E_0	0	4	6	7
E_1	1	5	4	8
E_2	2	6	5	9
E_3	3	8	7	9
E_4	4	8	5	10
E_5	5	8	9	10

Sub-tetrahedron	Nodes			
E_6	9	8	7	10
E_7	7	8	4	10
E_8	4	5	6	10
E_9	5	9	6	10
E_{10}	9	7	6	10
E_{11}	7	4	6	10

Table 4-1. Connectivity for the twelve sub-tetrahedra.

5. ADAPTIVE STRATEGY

5.1. OVERVIEW

In this work, we propose an 'intermediate' approach towards performing fully higher-order mesh adaption with the previously described composite tetrahedral element. Robust, reliable, and efficient mesh modification strategies exist for both isotropic and anisotropic adaptivity with linear simplical elements [12, 1, 18]. However, extensions of these techniques to higher-order tetrahedra still remains an open research topic [26, 10]. For this reason, we have chosen an approach whereby we leverage the robustness and reliability of linear mesh adaptation procedures at the expense of discarding some information about edge nodes produced by the solid mechanics finite element solve.

Our adaptive simulation strategy begins by generating a composite tetrahedral mesh that conforms to a desired geometry. A solid mechanics simulation is then initialized using the generated mesh with appropriate initial conditions and boundary conditions. The simulation is then run until a user-specified criterion triggers mesh adaptation. At this point, the mesh of composite tetrahedra contains edges that are potentially 'kinked', i.e. the location of an edge node in the mesh is not guaranteed to lie halfway between the edge's endpoint nodes.

As a first step, the reference configuration is updated right before every mesh adaptation occurs. The reference configuration is set to be the configuration given by the current deformed mesh. This allows us to perform solution mapping on quantities that define the reference configuration, thereby resetting the reference configuration on the final adapted mesh.

A linear tetrahedral representation of this deformed mesh is then obtained by omitting the positions of all edge nodes. This linear mesh, along with a *mesh size field* is passed to a linear tetrahedral mesh adaptation software called Omega_h [18]. For isotropic mesh adaptation, which we presently consider, the size field defines desired edge lengths at each vertex in the input mesh. The mesh adaptation software then performs a sequence of edge splits, swaps, and collapses in local cavities of elements to produce an adapted mesh of linear tetrahedra that closely satisfies the desired size field while also providing certain guarantees about the quality of every element in the adapted mesh (see Section 5.3).

To convert the adapted linear mesh back into a higher-order mesh of composite tetrahedra, the location of edge nodes must be specified for all edges in the mesh. To retain higher-order information from the solid mechanics solve, we first consider edges which exist both in the input linear mesh and the output (adapted) linear mesh. Since these edges existed prior to adaptation, we can retain the locations of their edge nodes prior to adaptation, as given by

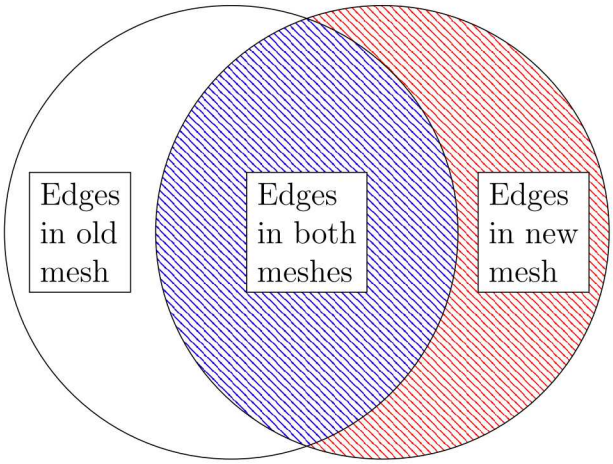


Figure 5-1. Venn-diagram of edges in the old (non-adapted) mesh and new (adapted) mesh. Edges in the white portion are no longer exist, edges in the blue portion remain kinked, and edges in the red portion (newly created edges) are assumed to be straight with a node at the edge's midpoint.

the solid mechanics solver. These edges are graphically represented in Figure 5-1 by the color blue. In the adapted mesh, the remaining edges without a specified node location are the edges that were created during mesh adaptation. These edges are represented in Figure 5-1 by the color red. For these remaining (new) edges, we place the node at the edge's midpoint, creating straight sided edges.

The process of forming the higher-order mesh can result in composite tetrahedral elements that have negative characteristic lengths (h) for time step purposes. Such elements are not suitable to perform numerical simulation. To rectify this, we straighten the edges of any such elements until a valid composite mesh is obtained. As a consequence, this further discards known information about edge nodes.

Next, both nodal and quadrature point solution information is transferred from the input mesh to the adapted mesh via a simple solution transfer scheme, although we propose a more sophisticated transfer scheme for future consideration. The solve/adapt cycle is then repeated until the final user-specified simulation time is reached.

We call this process an 'intermediate' approach because we have disregarded all higher-order shape information of the composite tetrahedra during the mesh adaptation process itself, but have still obtained an adapted composite tetrahedral mesh via post-processing. A more mathematically sound mesh adaptation process would evaluate the quality of composite tetrahedral elements directly taking into account all 10 nodes and would place new edge nodes at a location informed by the true quality of adjacent elements. In the remaining sections of this chapter, we discuss the details of our intermediate adaptive strategy.

5.2. REFERENCE CONFIGURATION UPDATES

Before any mesh adaptation occurs, we perform an update of variables that define the reference configuration. This, coupled with solution mapping during mesh adaptation, provides an updated reference configuration on the adapted mesh. To update the reference configuration, we set the reference density to be the density at the current time step n, we reset the deformation gradient to be the identity tensor I, and we update the initial deformation gradient F_{init} to be the accumulation of the deformation gradient at time step n and the initial deformation gradient $\boldsymbol{F}^b_{\text{init}}$ used until the time step n. This process is outlined by equation 5.1.

$$\begin{aligned} \rho_0 &= \rho^n \\ \boldsymbol{F} &= \boldsymbol{I} \\ \boldsymbol{F}_{\text{init}} &= \boldsymbol{F}^n \boldsymbol{F}^n_{init} \end{aligned} \tag{5.1}$$

Note, with the approach, we must use the total deformation gradient $\boldsymbol{F}_{\text{total}} := \boldsymbol{F}^n \boldsymbol{F}_{init}$ when calls to constitutive models are called.

5.3. TRIGGERING MESH ADAPTATION

To trigger mesh adaptation, we adopt an approach driven by element quality. While this approach is relatively simple, we emphasize that the criterion to trigger mesh adaptation can be as sophisticated as needed, and could even include information provided by error estimation routines.

A first approach to trigger mesh adaptation is to compute quality measures over all elements and adapt the mesh if the minimum quality over all elements is below some user-defined threshold. In our current implementation, we consider a quality measure over linear tetrahedra. Current efforts are developing quality measures specifically targeting the composite tetrahedral element, but these metrics are still in their infancy. Consequently, we utilize the so-called mean ratio element quality measure [25, 8, 24]. In equation 5.2, we define the mean ratio and note its interpretation as comparing the volume of linear tetrahedron to the volume of an equilateral tetrahedron with the same root-mean-squared edge length,

$$\eta_k = \left(\frac{V_k}{\gamma_k (\bar{l}_k)^3} \right)^{\frac{2}{3}}. \tag{5.2}$$

Here, V_k is the volume of the tetrahedron corresponding to element k, $\bar{l}_k$ is the root-mean-squared edge length of the tetrahedron, given as

$$\bar{l}_k := \left(\frac{1}{6} \sum_{i=0}^{6} (l_{k,i})^2 \right)^{\frac{1}{2}}, \tag{5.3}$$

where $l_{k,i}$ denotes the length of edge i in the tetrahedron, and γ_k is the volume of a equilateral tetrahedron with unit edge length,

$$\gamma_k := \frac{1}{\sqrt{72}}. \tag{5.4}$$

Note that the element quality measure satisfies the property $0 \leq \eta_k \leq 1$ for valid tetrahedron.

To determine whether or not mesh adaptation should occur, the minimum mean ratio element measure is computed over elements as

$$\eta_{k,\min} := \min_k \eta_k, \tag{5.5}$$

and then compared to two user-defined parameters: a minimum *desired* quality $\eta_{k,\min}^{\text{desired}}$ and a minimum *allowed* quality $\eta_{k,\min}^{\text{allowed}}$, where $\eta_{k,\min}^{\text{desired}} > \eta_{k,\min}^{\text{allowed}}$, such that adaptation occurs if

$$\eta_{k,\min} < \eta_{k,\min}^{\text{desired}} \quad \text{AND} \quad \begin{cases} \eta_{k,\min} \leq \eta_{k,\min}^{\text{old}} - 0.02 & \text{OR} \\ \eta_{k,\min} \leq \eta_{k,\min}^{\text{allowed}} + 0.02 \end{cases} \tag{5.6}$$

where $\eta_{k,\min}^{\text{old}}$ denotes the minimum mean ratio element measure at the previous time step. Thus, adaptation occurs if the element quality is deemed to be too poor at the current time step or if the element quality is deemed to be decreasing at an unsustainable rate.

5.4. MESH SIZE FIELDS

The input to the mesh adaptation software OMEGA_H is a so-called *mesh size field* that specifies a desired edge length h^{desired} at each mesh vertex in the mesh of linear tetrahedra. In the results below, we consider two mesh size field specifications, one which attempts to maintain element size throughout the simulation, and one that performs refinement with increasing equivalent plastic strain.

5.4.1. Implied Isotropic Size Field

To maintain element size throughout a simulation, we utilize a size field specification at each vertex v that averages the edge lengths h_{edge} of all edges adjacent to v, such that

$$h_v^{\text{iso}} = \frac{\sum_{edge=1}^{n_{\text{adj}}} h_{\text{edge}}}{n_{\text{adj}}}, \tag{5.7}$$

where n_{adj} denotes the number of edges adjacent to the vertex v.

This size field specification is particularly useful in the context of explicit Lagrangian finite element simulations in that the minimum characteristic element size (h) dictates the

maximum stable time step. In particular, as element sizes tend to zero, computational time will tend towards infinity. For instance, in a simulation with an inward converging geometry, such as the canonical Noh [32] test case, elements near the origin become increasingly compressed as simulation time increases. Without mesh adaptation, running such a test case to long times becomes intractable. This size field specification attempts to avoid this pitfall by preserving element sizes, thus preserving computational feasibility.

5.4.2. Refinement with Equivalent Plastic Strain

As a step towards adaptivity informed by error estimation information, we also consider a size field that performs refinement with increasing equivalent plastic strain. We consider three user-defined inputs: a minimal and maximal desired mesh size, $h_{\min}, h_{\max}$, respectively, and a maximal equivalent plastic strain $\epsilon_{p,\max}$. We set the mesh size field according to the formula:

$$h_v^{\text{eqps}} = \begin{cases} h_{\max} - \dfrac{(h_{\max} - h_{\min})}{\epsilon_{p,\max}} \bar{\epsilon}_{p,v} & \text{if} \quad \bar{\epsilon}_{p,v} \leq \epsilon_{p,max} \\ h_{\min} & \text{otherwise} \end{cases} \tag{5.8}$$

such that the mesh size h_v at a given vertex v decreases linearly with equivalent plastic strain ϵ_p until the maximal user-input equivalent plastic strain $\epsilon_{p,\max}$ is reached, at which point the mesh size field is held constant at the user defined minimal mesh size $h_{\min}$. In this manner, we achieve increased resolution in areas of the mesh with localized equivalent plastic strain while retaining element sizes that ensure computational feasibility. Figure 5-2 graphically illustrates this size field as a function of equivalent plastic strain.

Here, $\bar{\epsilon}_{p,v}$ denotes the average equivalent plastic strain at a given vertex, as computed as the average over values at n_{qps} integrations points in n_{adj} elements adjacent to the vertex v.

$$\bar{\epsilon}_{p,v} = \frac{\sum_{elem=1}^{n_{\text{adj}}} \sum_{qp=1}^{n_{\text{qps}}} \epsilon_{p,elem}\big|_{qp}}{n_{\text{adj}} n_{\text{qps}}}. \tag{5.9}$$

5.5. CONFORMAL MESH ADAPTIVITY WITH CAVITY OPERATIONS

Once a mesh size field has been specified, Omega_h performs local mesh modification operations over *cavities* of elements. The three mesh modification procedures we consider are edge splits, swaps, and collapses. For the sake of simplicity, Figure 5-3 demonstrates each of the three mesh modification procedures with linear triangular elements. These techniques readily extend to linear tetrahedral elements.

The edge split takes a given edge and places a new vertex at the edge's midpoint. New mesh edges are then created by connecting this new vertex to every other vertex in the cavity. For

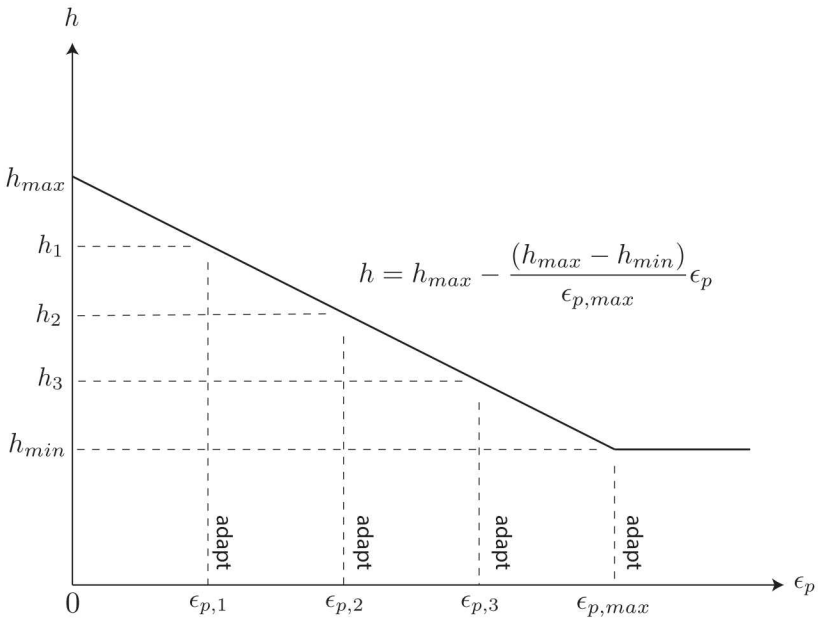


Figure 5-2. Size field function that decreases mesh size h with increasing equivalent plastic strain ϵ_p.

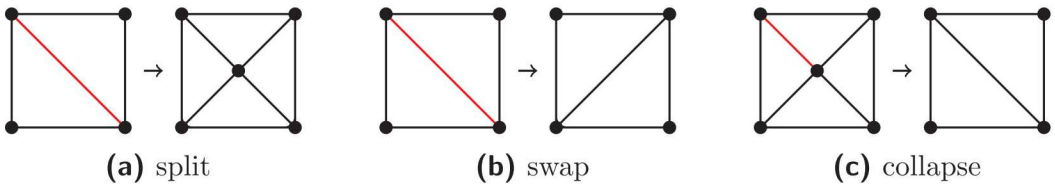


Figure 5-3. Split, swap, and collapse operations in two dimensions. Red indicates the edge the operation is acting upon.

instance, in Figure 5-3, a single edge split creates four new additional edges. Edge splitting is useful to *refine* the mesh to capture additional mesh resolution.

The edge swap takes a given mesh edge in cavity and creates a new mesh edge in the cavity by 'swapping' the mesh vertices that define the edge. For triangular elements, there is only one choice of newly created edge, as shown in Figure 5-3, however, in three dimensions, the choice of newly created edge is non-unique. The swap operation is typically used to improve mesh quality, and so we always choose the edge that leads to the highest quality elements in the cavity.

The edge collapse takes a chosen mesh edge and 'collapses' it by treating its two end vertices as one and the same. In general, the collapse operation is utilized to *coarsen* the mesh. For example, Figure 5-3 illustrates a collapse that results in two larger elements from four finer elements.

As mentioned previously, OMEGA_H performs sequences of these operations to satisfy the chosen mesh size field, as given in Section 5.4, but will also provide guaranteed lower bounds

on the minimum element quality. For our purposes, we utilize $\eta_k >= 0.3$, where η_k is given by equation 5.2.

5.6. EDGE STRAIGHTENING

Once a new valid linear tetrahedral mesh has been generated by OMEGA_H, edge nodes that exist in the intersection of the old mesh and new mesh are given their previous values, as determined by the solid mechanics solve. Additionally, the edge node for newly created edges is always placed at halfway between the vertex nodes connected to the edge. Figure 5-1 shows graphically how edge node positions are assigned.

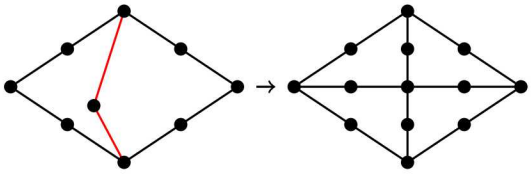


Figure 5-4. Splitting of an edge (red) for a 6-noded composite triangular element that yields well-defined resultant elements

As in section 5.5, we again consider triangular elements to clearly illustrate concepts, while noting that these concepts also extend to tetrahedral elements. Here, however, we consider 6-noded triangular elements that can have kinked edges, rather than straight-sided linear triangles.

In certain scenarios, the restriction that all newly created edges are straight-sided actually improves element quality. For example, Figure 5-4 demonstrates an edge split for a kinked edge that results in four well-shaped 6-noded triangles from a cavity that initially had two fairly deformed 6-noded triangles. This element quality comes at the cost of additional solution mapping, however, which may lead to an imperfect transfer of information. In contrast, a fully higher-order adaptive scheme may leave the red edge in Figure 5-4 kinked and place a new vertex at the red edge node, eliminating the need for solution mapping at that point.

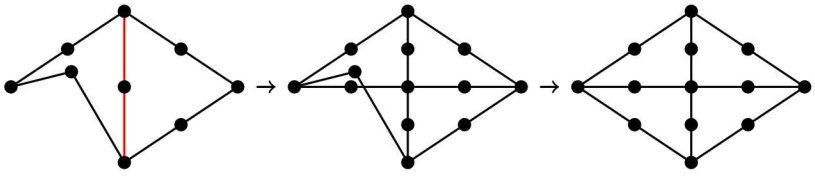


Figure 5-5. Splitting of an edge (red) for a 6-noded composite triangular element that yields an inverted resultant element and subsequent straightening of all edges adjacent to the inverted element

In other scenarios, however, newly created straight-sided edges can actually produce inverted elements. Figure 5-5 illustrates how this might occur with a split operation for an edge adjacent to a kinked element. To rectify inverted elements, we propose one of two strategies.

The first is to simply straighten all edges in the mesh adjacent to inverted elements as shown in the right-most image of Figure 5-5. This process must be performed iteratively, however, as straightening the edge in Figure 5-5 could result in an inverted element outside of the considered cavity.

As a more heavy-handed approach, we can also simply straighten all edges in the mesh. This approach is guaranteed to provide elements of good quality as it will simply yield the linear tetrahedral mesh produced by OMEGA_H with edge nodes placed at the edge's midpoint.

5.7. SOLUTION MAPPING

The transfer of field data from one mesh to another is a need that arises frequently within the context of mesh adaption in the finite element method [34, 36, 37, 40, 19, 33, 6]. Fields that are available at the nodes may be directly mapped by using the corresponding interpolation functions. The situation is more complicated, however, in simulations that carry state information in internal variables, as these are normally available only at integration points.

It is common for finite deformation constitutive models to be endowed with internal variables that do not belong to linear spaces. Nevertheless, a problem that is prevalent to the mapping procedures is that they transfer internal variables with the addition operator, typically via Lagrange interpolation. This operator, however, may not be admissible in the spaces in which these variables exist, and as a consequence such procedures do not, in general, guarantee that transferred internal variables remain in their appropriate spaces. For example, a scalar isotropic damage parameter might be extrapolated outside of its admissible range, $[0, 1)$.

5.7.1. Introduction

We advance a three-field finite element formulation as a method for recovering internal variables. The additional fields in the formulation are the *target* field of internal variables and a Lagrange multiplier that enforces the equality between the *source* and target internal variables. The Lagrange multiplier is subsequently identified as the corresponding conjugate thermodynamic force. The formulation leads to an expression for the target internal variables that is an L_2 projection of the source internal variables onto the space spanned by the interpolation functions selected for the extra fields. In effect, the projection extends the field of internal variables from the integration points to the entire domain. Once extended in this manner, values for the internal variables may be computed at any arbitrary point as a means of transferring the variables to a different mesh. By using a variational approach, the distance between the source and target internal variable fields is minimized in the L_2 norm of their space, and thus the projection is orthogonal and the corresponding operator is self-adjoint. The matrices that result from the operator are symmetric positive definite, sparse and of narrow band.

Next it is shown that for a set of common internal variables, the Lie algebras corresponding to these variables lie in linear spaces and are therefore suitable for polynomial interpolation.

The proposed recovery scheme preserves the constraints imposed on the internal variables as long as they belong to a Lie group. We adhere to the classical definition of internal variables as fields that are endowed with their own evolution equations as to eliminate the explicit dependence of the stored energy function from the history of deformation [31, 2]. Thus, the proposed variational principle states that thermodynamic forces conjugate to the internal variables (such as stresses) are not mapped, but are computed instead based on the target internal variables, thereby satisfying their own constraints (such as that stresses be within the elastic domain) and balance laws.

Assuming that internal variables belong to Lie groups, the corresponding Lie algebras may be obtained using the logarithmic map. Upon interpolation in the Lie algebra, the internal variable of interest is recovered by using the exponential map [27, 42, 20, 13]. A fast and accurate method for the computation of the exponential map is provided by the scaling and squaring algorithm with Padé approximants [16, 17]. Most tensor fields of interest, however, admit a polar decomposition and thus it proves convenient to interpolate the rotational and stretch components separately. An added benefit of this method is that the computation of logarithmic maps for these components is relatively straightforward.

Our previous work on mapping and updating internal variables via L_2 projections and Lie group interpolations did not contemplate the case of extremely large deformation [30]. When very large deformation is involved, the deformation gradient $\boldsymbol{F}$ may have a rotation component that has an angle of rotation $\theta \approx n\pi$, where $n \in \mathbb{Z}_{\geq 0}$. In that situation, algorithms to compute the principal logarithm of $\boldsymbol{F}$ may fail or require the use of complex arithmetic. This is undesirable. On the other hand, Mota et al. [30] advocate the use of the polar decomposition of the deformation gradient $\boldsymbol{F} = \boldsymbol{R}\boldsymbol{U}$, compute the logarithm of the rotation $\boldsymbol{R}$ and stretch $\boldsymbol{U}$ separately, map, and then reconstitute the deformation gradient field by means of the exponential map. This method requires the use of five tensor algorithms: polar decomposition; logarithmic map for rotations; logarithmic map for symmetric positive definite tensors; exponential map for symmetric tensors; and the exponential map for rotations. The use of specialized algorithms for rotations and symmetric tensors seeks to exploit their structure to minimize floating point operations.

5.7.2. Internal Variable Recovery

The formulation of our internal variable recovery technique may be carried out in the absence of dynamical effects inherent to the formulation of Chatper 3. Without loss of generality, we can restrict developments neglecting any inertial terms. introduce the energy functional

$$\Phi_0[\boldsymbol{\varphi}] := \int_\Omega A(\boldsymbol{F}, \boldsymbol{z}) \ \mathrm{d}V - \int_\Omega R\boldsymbol{B} \cdot \boldsymbol{\varphi} \ \mathrm{d}V - \int_{\partial_{\boldsymbol{T}}\Omega} \boldsymbol{T} \cdot \boldsymbol{\varphi} \ \mathrm{d}S, \tag{5.10}$$

in which $A(\boldsymbol{F}, \boldsymbol{z})$ is the Helmholtz free-energy density and $\boldsymbol{z}$ is a collection of internal variables.

5.7.2.1. Three-Field Functional

The functional 5.10 is modified by introducing a constraint as

$$\Phi[\boldsymbol{\varphi}, \bar{\boldsymbol{z}}, \bar{\boldsymbol{y}}] := \int_\Omega A(\boldsymbol{F}, \bar{\boldsymbol{z}}) \ \mathrm{d}V + \int_\Omega \bar{\boldsymbol{y}} \cdot (\bar{\boldsymbol{z}} - \boldsymbol{z}) \ \mathrm{d}V - \int_\Omega R\boldsymbol{B} \cdot \boldsymbol{\varphi} \ \mathrm{d}V - \int_{\partial_T B} \boldsymbol{T} \cdot \boldsymbol{\varphi} \ \mathrm{d}S, \quad (5.11)$$

in which $\bar{\boldsymbol{z}}$ is the field of *target* internal variables that is constrained to be equal to the *source* internal variables $\boldsymbol{z}$ by means of the Lagrange multiplier $\bar{\boldsymbol{y}}$. Although the Helmholtz free-energy density A is now evaluated using $\bar{\boldsymbol{z}}$ instead of $\boldsymbol{z}$, the functionals in 5.10 and 5.11 are equivalent at this stage due to this constraint.

Assume that $\boldsymbol{\varphi} \in U := (W_2^1(B))^3$, $\bar{\boldsymbol{z}}, \bar{\boldsymbol{y}} \in V := (W_2^1(B))^q$, in which $W_2^1(B)$ is the Sobolev space of square-integrable functions with square-integrable first derivatives, and q is the number of scalar parameters used to represent the collection of internal variables. The functional 5.11 is optimized by applying variations with respect to the independent fields $\boldsymbol{\varphi}$, $\bar{\boldsymbol{z}}$ and $\bar{\boldsymbol{y}}$. Define test functions corresponding to these fields as $\boldsymbol{\xi} \in U$, $\boldsymbol{\eta}, \boldsymbol{\zeta} \in V$, with $\boldsymbol{\xi} = \boldsymbol{0}$ on $\partial_{\boldsymbol{\varphi}}\Omega$. The variations follow as

$$D\Phi[\boldsymbol{\varphi}, \bar{\boldsymbol{z}}, \bar{\boldsymbol{y}}](\boldsymbol{\xi}) = \int_\Omega \boldsymbol{P} : \operatorname{Grad} \boldsymbol{\xi} \ \mathrm{d}V - \int_\Omega R\boldsymbol{B} \cdot \boldsymbol{\xi} \ \mathrm{d}V - \int_{\partial_T B} \boldsymbol{T} \cdot \boldsymbol{\xi} \ \mathrm{d}S = 0, \quad (5.12)$$

$$D\Phi[\boldsymbol{\varphi}, \bar{\boldsymbol{z}}, \bar{\boldsymbol{y}}](\boldsymbol{\eta}) = \int_\Omega (\bar{\boldsymbol{y}} - \boldsymbol{y}) \cdot \boldsymbol{\eta} \ \mathrm{d}V = 0, \quad (5.13)$$

$$D\Phi[\boldsymbol{\varphi}, \bar{\boldsymbol{z}}, \bar{\boldsymbol{y}}](\boldsymbol{\zeta}) = \int_\Omega (\bar{\boldsymbol{z}} - \boldsymbol{z}) \cdot \boldsymbol{\zeta} \ \mathrm{d}V = 0, \quad (5.14)$$

where $\boldsymbol{P} := \partial A / \partial \boldsymbol{F}$ is the first Piola-Kirchhoff stress and $\boldsymbol{y} := -\partial A / \partial \bar{\boldsymbol{z}}$ is the thermodynamic force conjugate to $\bar{\boldsymbol{z}}$. The corresponding Euler-Lagrange equations are

$$\begin{aligned} \operatorname{Div} \boldsymbol{P} + R\boldsymbol{B} = \boldsymbol{0} \text{ in } \Omega, \quad \boldsymbol{P}\boldsymbol{N} = \boldsymbol{T} \text{ on } \partial_T\Omega, \\ \bar{\boldsymbol{y}} = \boldsymbol{y} \text{ in } \Omega, \\ \bar{\boldsymbol{z}} = \boldsymbol{z} \text{ in } \Omega, \end{aligned} \quad (5.15)$$

as expected. Note that the equilibrium condition 5.15 is evaluated using the target internal variable field $\bar{\boldsymbol{z}}$, and therefore equilibrium and constitutive constraints are satisfied using this field. Next, introduce discretizations for the fields and test functions as

$$\begin{aligned} \boldsymbol{\varphi}_h(\boldsymbol{X}) &:= N_a(\boldsymbol{X})\boldsymbol{\varphi}_a \in U_h, \quad & \boldsymbol{\xi}_h(\boldsymbol{X}) &:= N_b(\boldsymbol{X})\boldsymbol{\xi}_b \in U_h, \\ \bar{\boldsymbol{z}}_h(\boldsymbol{X}) &:= \lambda_\alpha(\boldsymbol{X})\bar{\boldsymbol{z}}_\alpha \in V_h, \quad & \boldsymbol{\eta}_h(\boldsymbol{X}) &:= \lambda_\beta(\boldsymbol{X})\boldsymbol{\eta}_\beta \in V_h, \\ \bar{\boldsymbol{y}}_h(\boldsymbol{X}) &:= \lambda_\alpha(\boldsymbol{X})\bar{\boldsymbol{y}}_\alpha \in V_h, \quad & \boldsymbol{\zeta}_h(\boldsymbol{X}) &:= \lambda_\beta(\boldsymbol{X})\boldsymbol{\zeta}_\beta \in V_h, \end{aligned} \quad (5.16)$$

where N_a and N_b are interpolation functions for $\boldsymbol{\varphi}$ and $\boldsymbol{\xi}$, λ_α and λ_β are interpolation functions for $(\bar{\boldsymbol{z}}, \bar{\boldsymbol{y}})$ and $(\boldsymbol{\eta}, \boldsymbol{\zeta})$, and $(a, b) \in [1 \dots N]$ and $(\alpha, \beta) \in [1 \dots M]$, in which N is the number of nodes for $\boldsymbol{\varphi}$, M is the number of nodes for $\bar{\boldsymbol{z}}$ and $\bar{\boldsymbol{y}}$ respectively. $U_h \subset U$ and

$V_h \subset V$ are finite-dimensional subspaces of U and V respectively. Introducing these discretizations into the variational statements 5.12-5.14 gives

$$\int_\Omega \boldsymbol{P} \cdot \operatorname{Grad} N_a \ \mathrm{d}V - \int_\Omega R\boldsymbol{B} N_a \ \mathrm{d}V - \int_{\partial_T B} \boldsymbol{T} N_a \ \mathrm{d}S = \boldsymbol{0}, \tag{5.17}$$

$$\bar{\boldsymbol{y}}_h = \lambda_\alpha \left(\int_\Omega \lambda_\alpha \lambda_\beta \boldsymbol{I} \ \mathrm{d}V \right)^{-1} \int_\Omega \lambda_\beta \boldsymbol{y} \ \mathrm{d}V, \tag{5.18}$$

$$\bar{\boldsymbol{z}}_h = \lambda_\alpha \left(\int_\Omega \lambda_\alpha \lambda_\beta \boldsymbol{I} \ \mathrm{d}V \right)^{-1} \int_\Omega \lambda_\beta \boldsymbol{z} \ \mathrm{d}V, \tag{5.19}$$

which are the discrete statements of equilibrium, the discrete thermodynamic forces and discrete target internal variables, respectively, and with $\boldsymbol{I}$ being the $q \times q$ identity. Note that λ_α and λ_β form a basis for the space V_h, therefore 5.18 and 5.19 are projections of the fields $\boldsymbol{y}$ and $\boldsymbol{z}$ onto V_h. Effective computation of the integrals in these expressions readily suggests the use of interpolation functions λ_α and λ_β that are of the same order or less than N_a and N_b. This allows the application of the same integration scheme for 5.18 and 5.19 as for 5.17, ensuring that the projections will be fully integrated (provided that full integration is used for the equilibrium statement as well) using the same integration points, thus avoiding any unnecessary transfer of variables at this stage. Furthermore, if the interpolation functions λ_α and λ_β are discontinuous across element boundaries, the projection 5.19 reduces to extrapolation of values from integration points to nodes (cf. Ortiz and Quigley, Radovitzky and Ortiz). For convenience, introduce

$$\boldsymbol{H}_{\alpha\beta} := \int_\Omega \lambda_\alpha \lambda_\beta \boldsymbol{I} \ \mathrm{d}V, \quad \boldsymbol{\Sigma}_\beta := \int_\Omega \lambda_\beta \boldsymbol{z} \ \mathrm{d}V, \quad \bar{\boldsymbol{z}}_\alpha := \boldsymbol{H}^{-1}_{\alpha\beta} \boldsymbol{\Sigma}_\beta, \tag{5.20}$$

in which $\bar{\boldsymbol{z}}_\alpha$ are the nodal values of the discrete target field. Then the projection 5.19 can also be expressed as

$$\bar{\boldsymbol{z}}_h(\boldsymbol{X}) = \lambda_\alpha(\boldsymbol{X}) \bar{\boldsymbol{z}}_\alpha. \tag{5.21}$$

Herein, we advocate the use of the singular value decomposition (SVD) to address the case of very large deformation. An additional benefit is the reduction of storage and decrease of the number of tensor algorithms from five to four, as follows. First compute the SVD of the deformation gradient $\boldsymbol{F} = \boldsymbol{X}\boldsymbol{D}\boldsymbol{Y}^{\mathrm{T}}$, then compute the polar decomposition of the deformation gradient $\boldsymbol{F} = \boldsymbol{R}\boldsymbol{U}$, where $\boldsymbol{R} = \boldsymbol{X}\boldsymbol{Y}^{\mathrm{T}}$ and $\boldsymbol{U} = \boldsymbol{Y}\boldsymbol{D}\boldsymbol{Y}^{\mathrm{T}}$, then compute the logarithmic map of each of the components $\boldsymbol{r} := \log \boldsymbol{R}$, and $\boldsymbol{u} := \log \boldsymbol{U} = \boldsymbol{Y} \log(\boldsymbol{D}) \boldsymbol{Y}^{\mathrm{T}}$. Then map, apply the exponential map, and reconstitute the mapped deformation gradient. Since $\boldsymbol{r}$ and $\boldsymbol{u}$ require the storage of three and six components correspondingly, the amount of storage is the same as for the deformation gradient $\boldsymbol{F}$ or its principal logarithm $\log \boldsymbol{F}$. The required logarithmic maps are those corresponding to either scalars or proper rotations, and the corresponding exponential maps are those of a rotation pseudo-vector and a symmetric tensor. This is described in detail in Algorithm 1.

Note that mapping the components $\boldsymbol{X}$, $\boldsymbol{D}$, and $\boldsymbol{Y}$ of the SVD separately leads to incorrect results. This is due to the non-uniqueness of the singular value decomposition, even for non-singular $\boldsymbol{F}$. Likewise, mapping $\boldsymbol{Y}$ separately as a rotation leads to incorrect results due

1: $t \leftarrow 0$, $\boldsymbol{F} \leftarrow \boldsymbol{I}$
2: **repeat** ▷ Time integration loop
3: Compute increment of deformation gradient $\triangle \boldsymbol{F}$
4: $\boldsymbol{F} \leftarrow \triangle \boldsymbol{F} \cdot \boldsymbol{F}$
5: **if** mapping required **then**
6: $[\boldsymbol{X}, \boldsymbol{D}, \boldsymbol{Y}] \leftarrow \mathrm{SVD}(\boldsymbol{F})$
7: $\boldsymbol{R} \leftarrow \boldsymbol{X}\boldsymbol{Y}^{\mathrm{T}}$
8: $\boldsymbol{r} \leftarrow \mathrm{RVofRT}(\boldsymbol{R})$
9: $\boldsymbol{s} \leftarrow \text{align-rvs}(\boldsymbol{r})$
10: $\boldsymbol{d} \leftarrow \mathrm{diag}(\log(\mathrm{diag}(\boldsymbol{D})))$
11: $\boldsymbol{u} \leftarrow \mathrm{symm}(\boldsymbol{Y}\boldsymbol{d}\boldsymbol{Y}^{\mathrm{T}})$ ▷ $\boldsymbol{U} = \boldsymbol{Y}\boldsymbol{D}\boldsymbol{Y}^{\mathrm{T}}$
12: Map $\boldsymbol{s}, \boldsymbol{u}$ using additive interpolation.
13: $\boldsymbol{R} \leftarrow \mathrm{RTofRV}(\boldsymbol{s})$
14: $\boldsymbol{U} \leftarrow \exp(\boldsymbol{u})$
15: $\boldsymbol{F} \leftarrow \boldsymbol{R}\boldsymbol{U}$
16: **end if**
17: Compute time step $\triangle t$
18: $t \leftarrow t + \triangle t$
19: **until** $t \geq T_{\text{final}}$

Algorithm 1 Mapping algorithm for the deformation gradient $\boldsymbol{F}$ via the SVD.

to its non-uniqueness when the stretch $\boldsymbol{U} = \boldsymbol{Y}\boldsymbol{D}\boldsymbol{Y}^{\mathrm{T}}$ has repeated eigenvalues. On the other hand, the components $\boldsymbol{R}$ and $\boldsymbol{U}$ of the polar decomposition are unique for a non-singular $\boldsymbol{F}$.

5.7.3. Tensor operators

The steps outlined in Algorithm 1 require the computation of several tensor functions and operators. The first is the computation of the rotation pseudo-vector from a rotation tensor in Line 8. This is accomplished in a singularity-free manner by using a quaternion parametrization of the rotations, and by recourse to Spurrier's algorithm [44]. A Julia[1] implementation of this operator is shown in Listing 5.1, which in turn relies on the code shown in Listing 5.2 and Listing 5.3. These algorithms were adapted from Balopoulos and Abel [3]. Note that the quaternion parametrization is not exposed to the user of these algorithms, and also that

$$\boldsymbol{r} := \mathrm{RVofRT}(\boldsymbol{R}) \equiv \operatorname{check} \log \boldsymbol{R}, \tag{5.22}$$

and

$$\boldsymbol{R} := \mathrm{RTofRV}(\boldsymbol{r}) \equiv \exp \operatorname{check} \boldsymbol{r}. \tag{5.23}$$

The check operator is defined in such a way that if applied to a vector $\boldsymbol{v} \in \mathbb{R}^3$, it returns a skew symmetric tensor $\check{\boldsymbol{v}} \in so(3)$, and if applied to a skew symmetric tensor $\check{\boldsymbol{v}} \in so(3)$, it returns a vector $\boldsymbol{v} \in \mathbb{R}^3$. In both cases, the operator is constructed such that

[1] Julia is available on: `https://julialang.org/`

Listing 5.1 RVofRT: rotation pseudo-vector of rotation tensor.

```
function RVofRT(R)
    q = QofRT(R)
    w = RVofQ(q)
    return w
end
```

$\check{\boldsymbol{v}} \cdot \boldsymbol{u} \equiv \boldsymbol{v} \times \boldsymbol{u} \; \forall \; \boldsymbol{u} \in \mathbb{R}^3$, that is

$$\boldsymbol{v} = \operatorname{check} \check{\boldsymbol{v}} \Leftrightarrow \check{\boldsymbol{v}} = \operatorname{check} \boldsymbol{v}, \quad \boldsymbol{v} = \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix}, \quad \check{\boldsymbol{v}} = \begin{pmatrix} 0 & -v_3 & v_2 \\ v_3 & 0 & -v_1 \\ -v_2 & v_1 & 0 \end{pmatrix}. \tag{5.24}$$

The diagonal operator is defined in a similar fashion as

$$\boldsymbol{v} = \operatorname{diag} \hat{\boldsymbol{v}} \Leftrightarrow \hat{\boldsymbol{v}} = \operatorname{diag} \boldsymbol{v}, \quad \boldsymbol{v} = \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix}, \quad \check{\boldsymbol{v}} = \begin{pmatrix} v_1 & 0 & 0 \\ 0 & v_2 & 0 \\ 0 & 0 & v_3 \end{pmatrix}. \tag{5.25}$$

The alignment function in Line 9 of Algorithm 1 is necessary for the interpolation of rotation pseudo-vectors. This is due to rotations that have an angle of rotation that is near π. In that case, the corresponding rotation tensors may be arbitrarily close to one another, yet interpolating their corresponding rotation pseudo-vectors yields incorrect results. Consider the rotation pseudo-vectors $\boldsymbol{a} = [\pi - \epsilon, 0, 0]$ and $\boldsymbol{b} = [\epsilon - \pi, 0, 0]$, where $\epsilon \ll 1$. These two pseudo-vectors have rotation tensors that are very close to one another. Yet directly interpolating the pseudo-vectors results in $\boldsymbol{c} := \frac{1}{2}(\boldsymbol{a} + \boldsymbol{b}) = [0, 0, 0]$, which is grossly incorrect. By contrast, if we re-align $\boldsymbol{b}$ by $\tilde{\boldsymbol{b}} := \boldsymbol{b} - 2\pi * \frac{\boldsymbol{b}}{||\boldsymbol{b}||} = [\pi + \epsilon, 0, 0]$, we obtain $\boldsymbol{c} := \frac{1}{2}(\boldsymbol{a} + \tilde{\boldsymbol{b}}) = [\pi, 0, 0]$, which is correct. Likewise, if we re-align $\boldsymbol{a}$ by $\tilde{\boldsymbol{a}} := \boldsymbol{a} - 2\pi * \frac{\boldsymbol{a}}{||\boldsymbol{a}||} = [-\pi - \epsilon, 0, 0]$, we obtain $\boldsymbol{c} := \frac{1}{2}(\tilde{\boldsymbol{a}} + \boldsymbol{b}) = [-\pi, 0, 0]$, which is also correct. A heuristic algorithm to perform this alignment is shown in Algorithm 2.

Finally, the computation of a rotation tensor from a rotation pseudo-vector is performed by means of Listing 5.4, which uses the algorithm often accredited to Rodrigues [13], as implemented in Listing 5.5, Listing 5.6, and Listing 5.7. This completes the description of the mapping algorithm for the deformation gradient for the case of very large deformation.

Listing 5.2 QofRT: quaternion of rotation tensor.

```
# This function implements the singularity-free algorithm due to
# Spurrier.
#
# First the scalar "maxm" is defined as the maximum among the diagonal
# terms and the trace of the rotation tensor.
#
# i)  If maxm is equal to the trace of R, then:
#       2 * qs = sqrt(1 + maxm)
#     and
#       qv = axial_vec(skew(R)) / (2 * qs)
#
# ii) If maxm is equal to R[j][j] (no summation), then:
#       2 * qv[j] = sqrt(2 * maxm + 1 - tr(R))
#       qs = axial_vec(skew(R))[j] / (2 * qv[j])
#     and for i,j,k all different:
#       qv[k] = off_diag_vec(symm(R))[i] / (2 * qv[j])
#
# Since the rotation tensor is a quadratic function of the quaternion,
# at least one square root has to be evaluated to get back the
# quaternion.  After that, only divisions are needed and the divisor
# should be bounded as far from zero as possible.
#
function QofRT(R)
    trR = tr(R)
    maxm = trR
    maxi = 4
    q = zeros(4)
    for i = 1 : 3
        if R[i, i] > maxm
            maxm = R[i, i]
            maxi = i
        end
    end
    if maxi == 4
        root = sqrt(maxm + 1.0)
        factor = 0.5 / root
        q[1] = 0.5 * root
        q[2] = factor * (R[3, 2] - R[2, 3])
        q[3] = factor * (R[1, 3] - R[3, 1])
        q[4] = factor * (R[2, 1] - R[1, 2])
    elseif maxi == 3
        root = sqrt(2.0 * maxm + 1.0 - trR)
        factor = 0.5 / root
        q[1] = factor * (R[2, 1] - R[1, 2])
        q[2] = factor * (R[1, 3] + R[3, 1])
        q[3] = factor * (R[2, 3] + R[3, 2])
        q[4] = 0.5 * root
    elseif maxi == 2
        root = sqrt(2.0 * maxm + 1.0 - trR)
        factor = 0.5 / root
        q[1] = factor * (R[1, 3] - R[3, 1])
        q[2] = factor * (R[1, 2] + R[2, 1])
        q[3] = 0.5 * root
        q[4] = factor * (R[2, 3] + R[3, 2])
    elseif maxi == 1
        root = sqrt(2.0 * maxm + 1.0 - trR)
        factor = 0.5 / root
        q[1] = factor * (R[3, 2] - R[2, 3])
        q[2] = 0.5 * root
        q[3] = factor * (R[1, 2] + R[2, 1])
        q[4] = factor * (R[1, 3] + R[3, 1])
    end
    return q
end
```

Listing 5.3 RVofQ: rotation pseudo-vector of quaternion.

```
# This function maps a quaternion, qq = (qs, qv), to its
# corresponding "principal" rotation pseudo-vector, aa, where
# "principal" signifies that |aa| <= π.  Both qq and -qq map into the
# same rotation matrix. It is convenient to require that qs >= 0, for
# reasons explained below.  The sign inversion is applied to a local
# copy of qq to avoid side effects.
#
#    |qv| = | sin(|aa| / 2) |
#    qs  =   cos(|aa| / 2)
#      <==>
#    |aa| / 2 = k * π (+ or -) asin(|qv|)
#    |aa| / 2 = 2 * l * π (+ or -) acos(qs)
#
# The smallest positive solution is:    |aa| = 2 * acos(qs)
# which satisfies the inequality:   0 <= |aa| <= π
# because of the assumption  qs >= 0.  Given |aa|, aa
# is obtained as:
#
#    aa = (|aa| / sin(acos(qs))) qv
#       = (|aa| / sqrt(1 - qs^2)) qv
#
# The procedure described above is prone to numerical errors when qs
# is close to 1, i.e. when |aa| is close to 0.  Since this is the most
# common case, special care must be taken.  It is observed that the
# cosine function is insensitive to perturbations of its argument in
# the neighborhood of points for which the sine function is conversely
# at its most sensitive.  Thus the numerical difficulties are avoided
# by computing |aa| and aa as:
#
#    |aa| = 2 * asin(|qv|)
#    aa  = (|aa| / |qv|) qv
#
# whenever qs  is close to 1.
#
function RVofQ(qq)
    if qq[1] >= 0
        q = qq
    else
        q = -qq
    end
    qs = q[1]
    qv = [q[2], q[3], q[4]]
    qvnorm = norm(qv)
    aanorm = 2.0 * (qvnorm < sqrt(0.5) ? asin(qvnorm) : acos(qs))
    coef = qvnorm < sqrt(eps()) ? 2.0 : aanorm / qvnorm
    aa = coef * qv
    return aa
end
```

Listing 5.4 RTofRV: rotation tensor of rotation pseudo-vector.

```
function RTofRV(w)
    q = QofRV(w)
    R = RTofQ(q)
    return R
end
```

Require: $n \geq 2$ ▷ Number of nodes in this element of source mesh.
Require: $\alpha \in (0, 1) \approx 1$ ▷ Tolerance for opposite pseudo-vectors mapping to $\approx \pi$ rotations.
function ALIGN-RVS($\boldsymbol{r}_i,\ i \in [1, n]$) ▷ Nodal rotation pseudo-vectors for interpolation.
  for $i \leftarrow 1, n$ **do**
    $\boldsymbol{s}_i \leftarrow \boldsymbol{r}_i$ ▷ Copy pseudo-vectors to avoid changing originals.
  end for
  $s \leftarrow ||\boldsymbol{s}_1||$
  for $i \leftarrow 2, n$ **do**
    if $\boldsymbol{s}_i \cdot \boldsymbol{s}_1 \leq -\alpha * \pi^2$ **then** ▷ Pseudo-vectors are nearly opposite with angle $\approx \pi$
      $\boldsymbol{s}_i \leftarrow \boldsymbol{s}_i - 2 * \pi * \frac{\boldsymbol{s}_i}{||\boldsymbol{s}_i||}$
    end if
    $s \leftarrow s + ||\boldsymbol{s}_i||$
  end for
  if $s > n\pi$ **then** ▷ Re-align so that $||\boldsymbol{s}_i|| \leq \pi$
    for $i \leftarrow 1, n$ **do**
      $\boldsymbol{s}_i \leftarrow \boldsymbol{s}_i - 2 * \pi * \frac{\boldsymbol{s}_i}{||\boldsymbol{s}_i||}$
    end for
  end if
  return $\boldsymbol{s}_i,\ i \in [1, n]$
end function

Algorithm 2 Alignment of rotation pseudo-vectors for interpolation.

Listing 5.5 QofRV: quaternion of rotation pseudo-vector.

```
# This function maps a rotation pseudo-vector, aa, to a quaternion, qq
# = (qs, qv), where qv is a vector and qs a scalar, defined as
# follows:
#
#   qv = sin(|aa| / 2) * aa / |aa|
#   qs = cos(|aa| / 2)
#
function QofRV(aa)
    halfnorm = 0.5 * norm(aa)
    temp = 0.5 * Ψ(halfnorm)
    qq = zeros(4)
    qq[2 : 4] = temp * aa
    qq[1] = cos(halfnorm)
    return qq
end
```

Listing 5.6 RTofQ: rotation tensor of quaternion.

```
# For qq = (qs, qv), where qv is a vector and qs is a scalar, one
# can write:
#
#    R = 2 * qv \otimes qv
#      + 2 * qs * check(qv)
#      + (2 * qs^2 - 1) * I
#
function RTofQ(qq)
    qs = qq[1]
    qv = [qq[2], qq[3], qq[4]]
    I = [1 0 0; 0 1 0; 0 0 1]
    R = 2.0 * qv * qv' + 2.0 * qs * check(qv) + (2.0 * qs * qs - 1.0) * I
    return R
end
```

Listing 5.7 $\Psi(x)$: singularity-free $\frac{\sin x}{x}$.

```
# In the algebra of rotations one often comes across functions that
# take undefined (0/0) values at some points.  Close to such points
# these functions must be evaluated using their asymptotic
# expansions; otherwise the computer may produce wildly erroneous
# results or a floating point exception.  To avoid unreadable code
# everywhere such functions are used, we introduce here functions to
# the same effect.
#
# NAME  FUNCTION FORM      X    ASSYMPTOTICS    FIRST RADIUS    SECOND RADIUS
# ----  -------------      -    ------------    ------------    -------------
# Ψ     sin(x)/x           0    1.0(-x^2/6)     (6*EPS)^.5      (120*EPS)^.25
#
# ΨXT
function Ψ(x)
    y = abs(x)
    e2 = sqrt(eps())
    e4 = sqrt(e2)
    if (y > e4)
        return sin(y) / y
    elseif (y > e2)
        return 1.0 - y * y / 6.0
    else
        return 1.0
    end
end
```

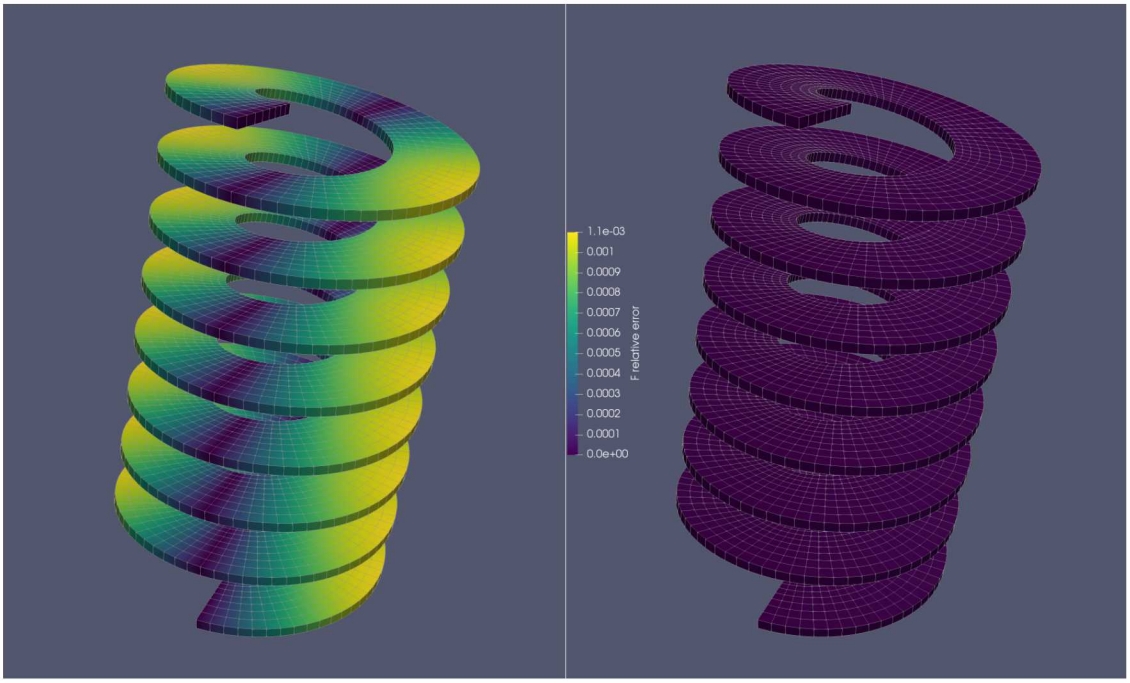


Figure 5-6. Relative error $||\boldsymbol{F}_{\text{target}} - \boldsymbol{F}_{\text{source}}||/||\boldsymbol{F}_{\text{source}}||$ of the magnitude of the deformation gradient for a straight bar deformed into a helix. The error is computed after mapping the deformation gradient $\boldsymbol{F}_{\text{source}}$ from a source mesh to a target mesh, resulting in a target deformation gradient $\boldsymbol{F}_{\text{target}}$. The meshes are designed to lack common nodes except at the eight corners of the domain, and only the target mesh is shown in the figure. The left helix shows the resultant error of using linear interpolation for mapping. The right helix shows the error obtained by using Lie group interpolation and the techniques advocated herein.

6. RESULTS

6.1. INTRODUCTION

Prior to illustrating proof-of-concept simulations of in-core mesh adaptivity, the team sought to establish a relatively strong theoretical basis through verification. We first examined the kinematics through carefully constructed deformation paths which were complemented by rigid body translations and rotations. Additional work confirmed the explicit time integration. The composite tetrahedral element was unit tested for the gradient operator, projected Jacobian (for integration), deformation gradient, volume-averaged Jacobian (of deformation), 1st Piola-Kirchhoff stress, volume-averaged pressure, residual, and mass matrix. To be clear, all the element unit tests implemented in the production environment, Sierra/SM, were replicated in LGR. We even added a new unit test for calculating the mass matrix with varying input density. First and foremost, we employ unit tests to speed the implementation of technology. These nightly tests also provided some assurance that our efforts to rapidly prototype mesh adaptivity would not impact the theoretical basis of our research environment.

Because real problems invoke constitutive models of complexity, we also sought to have common constitutive models for both the research environment and the production environment. To that end, it is important to emphasize that the J_2 model in LGR, `fefp`, is identical to the J_2 model in Sierra/SM. This is an essential ingredient for verification of the solutions obtained in LGR, as any differences in solutions with respect to Sierra/SM cannot be attributed to differences in material models. In addition, it is also noteworthy to mention that the introduction of the `fefp` plasticity model in LGR was greatly facilitated by code design. Essentially, the `fefp` code from Sierra/SM was ported almost as-is to LGR, with some very minor differences to account for their respective interfaces to material models.

Given the compacted time line of the effort and need to both align and illustrate progress applicable to the production environment, our team elected to compare solutions to relevant problems of increasing complexity in both Sierra/SM and LGR. Given very favorable comparisons without adaption, we concluded that the computational infrastructure in our research environment to be a suitable baseline upon which to build adaptive processes. Indeed, we illustrated extremely favorable comparisons for dynamic use cases rooted in both hyperelasticity and elastoplasticity. Convergence studies in in LGR confirm prior findings in the production environment. With the composite tetrahedral element, we can resolve large plastic deformations with accurate pressure distributions.

These targeted studies provide the requisite segue to in-core mesh adaption. Although we explored may themes, we converged on a few practical applications that have the potential to resonate with analysts.

1. Given my computational budget, can I invoke mesh adaptivity to resolve large deformations under constant element size? What is the best I can do given my current limitations?

2. I'm currently evaluating a multiplicity of loading environments. Although I have the ability to resolve localization, uniform refinement is not feasible. Can I employ mesh adaptivity for local refinement? Through local refinement, can I achieve a 10X decrease in the number of degrees of freedom?

The following sections illustrate a baseline for solution and the aforementioned practical themes for adaptivity.

6.2. TORSION OF A HYPERELASTIC BAR

Prior to inelasticity, we sought to confirm the element formulation and the central-difference explicit time integration. For this particular example, we employ a rectangular bar of length 6 with in-plane dimension 1. An element size of 0.2 placed 5 tetrahedral elements along each edge. To preserve regularity along the major axis, the bar was first discretized into hexahedral elements and then split (via CUBIT) into tetrahedral elements. In both code bases, we employed a decoupled Neohookean model for elasticity with a Young's Modulus of 2.83 GPa and a shear modulus of 0.567 GPa. To emphasize the dynamics of the process, an initial velocity of sinusoidal character is applied

$$\begin{aligned} v_1(0) &= 100 \sin\left(\frac{\pi}{12} X_2\right) X_3 \\ v_2(0) &= 0 \\ v_3(0) &= -100 \sin\left(\frac{\pi}{12} X_2\right) X_1. \end{aligned} \tag{6.1}$$

Both the research and production codes then integrate the equations of motion as the bars deform, spin down, stop, and then begin to spin up. Part of the evolutionary process is illustrated in Figure 6-1. A consistent projection of the effective stress is illustrated in SIERRA/SM. The wireframe of the LGR is overlaid for comparison. The deformed discretizations match over the entirety of the simulation. The comparison notes 4,000 time steps.

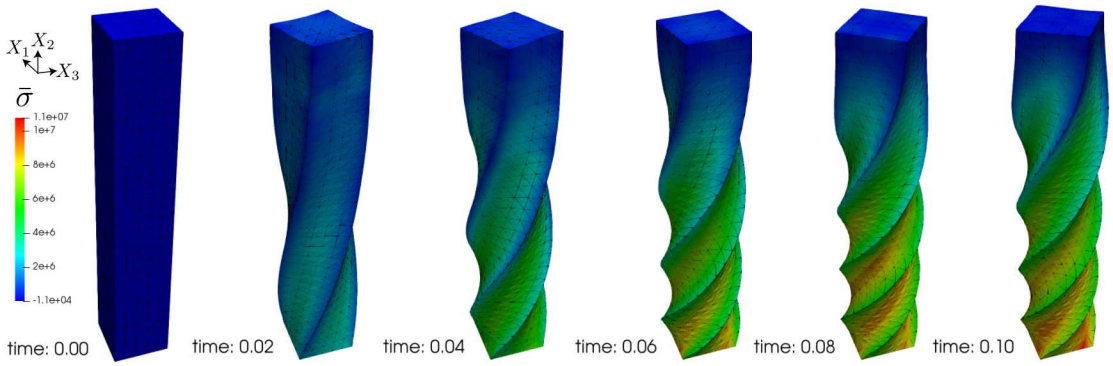


Figure 6-1. Torsion of a hyperelastic bar subjected to a sinusoidal initial velocity field. Comparison of the effective stress in SIERRA/SM and the wireframe in LGR for numerous deformed configurations. Both codes took approximately 4,000 time steps.

6.3. LARGE DEFORMATION PLASTICITY

We revisit the deformation of a circular-slotted, stainless steel bar illustrated in prior work [35]. Both the stress concentrations and the inherent hardening of 304L stainless steel provide a useful vehicle for illustrating adaptivity for large deformations. For reference, we illustrate an initial discretization of the geometry of interest in Figure 6-2. Our initial goal to to establish a baseline that mirrors prior work. We then seek to compare the research and production environments for large deformation plasticity. In prior work, we employed a piecewise-linear hardening model. To accommodate current modeling capabilities, we seek to employ a common functional form hardening, Voce hardening. Although multiple models could be explored, we elected to pursue the `fefp` model in SIERRA/SM for a clean partitioning of the kinematics, robustness, and hyperelasticity. As previously mentioned, the model was ported to LGR with few issues. The fit is illustrated in Figure 6-3. Clearly the new model fit has the same character as prior work. Note that substantial hardening is evident to equivalent plastic strains (eqps) of 2.0. This is indeed representative of austenitic stainless steels and promotes a stable localization process for the geometry of interest. We employed a Young's modulus and Poisson's ratio of 200 GPa and 0.249, respectively. For Voce hardening, the hardening modulus was 1.23 GPa with an exponential coefficient of 0.55. Having common material models and parameters in both SIERRA/SM and LGR, simulations were performed for the same mesh discretization. Although we can easily discretize the body to have mid-edge nodes be placed on curved boundaries, we elected to constrain the mid-edge nodes to the midpoint. Presently, LGR constructs tet10 configurations from tet4 configurations. A consistent comparison requires SIERRA/SM to employ the same tet4-derived discretization. A snapshot at large deformations is illustrated in Figure 6-4. In this case, the equivalent plastic strain has been volume averaged over each element in SIERRA/SM. The discontinuities evident in the heterogeneous fields are mirrored by the overlaid wireframe from LGR. We note this excellent agreement is captured at a common time approximately 230,000 steps into the simulation. Through this work and prior studies, we believe that the mechanics operable is viable and provides a baseline for studies

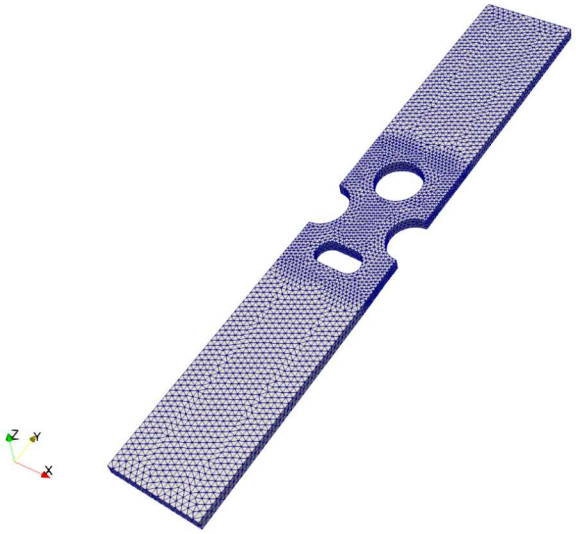


Figure 6-2. Tensile geometry having stresses concentrations subjected to far-field tension. The large deformation of stainless steel alloys motivates mesh adaptivity.

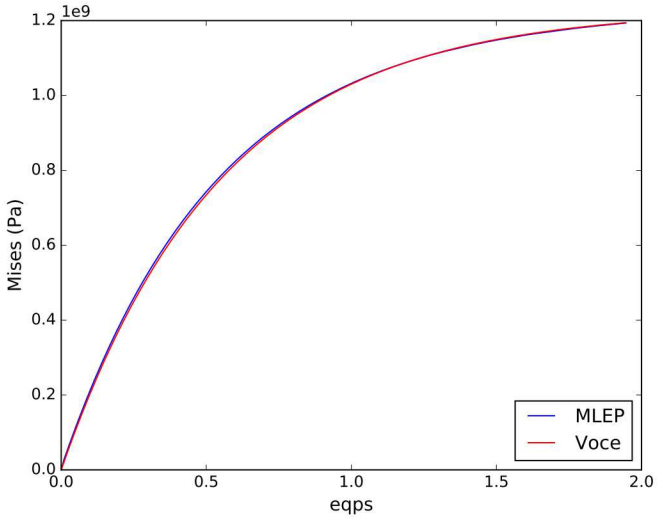


Figure 6-3. In an effort to align current efforts with past work, we parameterized Voce hardening parameters to sufficiently represent a piecewise-linear basis. Significant hardening a large equivalent plastic strains (eqps) promotes stability in the localization process.

in adaptivity.

For additional completeness, we pursued a mesh refinement study to illustrate convergence in the equivalent plastic strain and the pressure field at finite plastic strains in LGR. Figures illustrating convergence are shown in Figure 6-5 and Figure 6-6. For both figures, we illustrate the model system having symmetry across the width (x) and through the thickness (z). Because one cannot currently volume-average quantities at the element level in LGR, we

elect to display field quantities at the first integration point (0). In the case of pressure, all integration points yield the same pressure (through a volume-averaged Jacobian), so one can view Figure 6-6 as a volume average. The maximum plastic strains at times of 25 ms, 50 ms, and 75 ms were 0.48, 0.85, and 1.5, respectively. As noted Figure 6-5, the deformation is localizing to the surface of an evolving notch. The pressure field is less localized given the asymptotic nature of the hardening noted in Figure 6-3. Even integration point quantities are quite smooth with increasing mesh refinement. To be clear, the pressure in LGR is defined to be $-\text{tr}[\boldsymbol{\sigma}]/3$. The color scale is reversed to help the reader. Although this study is not exactly spellbinding, it does effectively illustrate that the composite tetrahedral element does not lock under severe plastic deformation. Specifically, those conclusions can be drawn from LGR for bodies subjected to explicit transient dynamics. We now embark on a series of studies to examine the promise of local adaptivity.

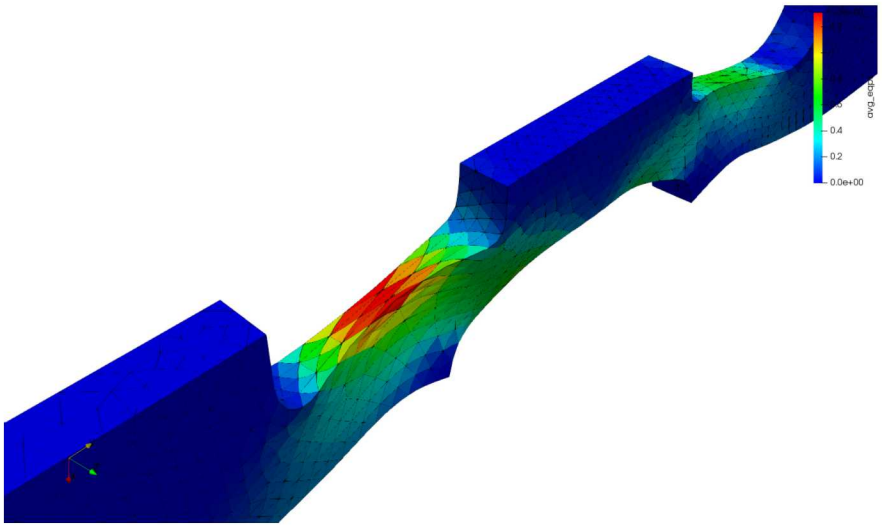


Figure 6-4. Comparison of the equivalent plastic strain in SIERRA/SM and the wireframe in LGR for large deformations. Both codes took approximately 4000 time steps.

6.4. RESOURCE DRIVEN ADAPTION

The first thought experiment revolves around finite resources and large deformations. In system-level models, we often seek to study phenomena for multiple environments. One's ability to resolve phenomena via mesh size often stems from the explicit time step. As elements deform in abnormal environments, the time step decreases. In order to keep the simulation viable, analysts will "kill" based on time step. Elements with explicit time steps below a threshold are removed from the simulation. If the goal is to resolve localization, we must accept that traditional workflows remove elements in the region of localization. What if, however, we could adaptively remesh the body to preserve the time step? What if, under large deformations, the time step remained essentially unchanged? We could then allocate resources and determine the best solution achievable with a particular level of resolution. We

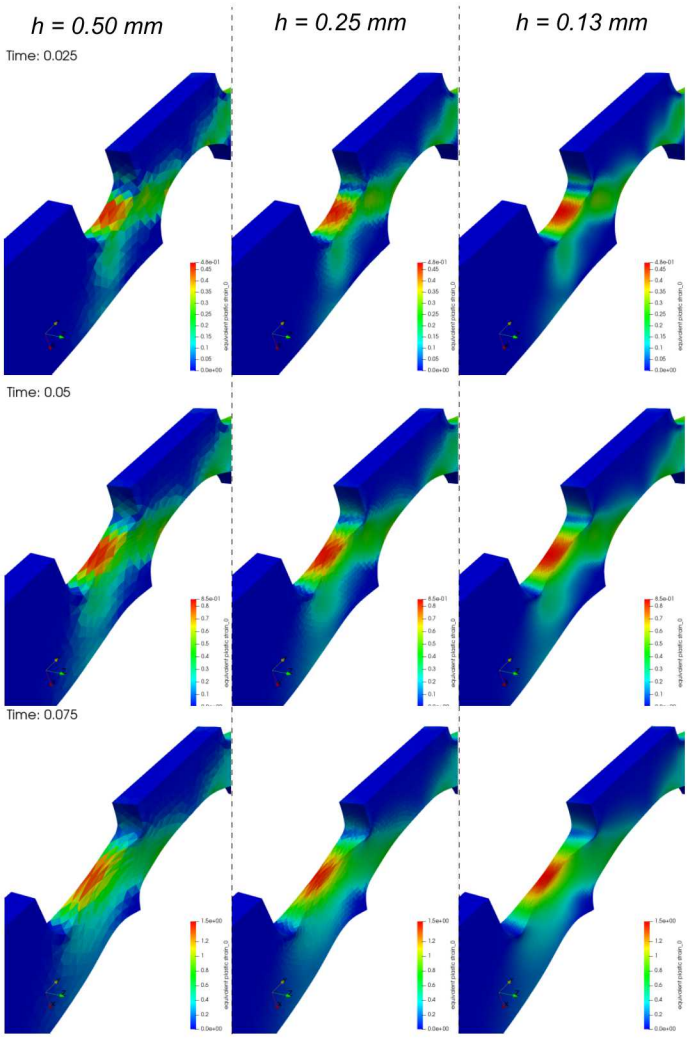


Figure 6-5. Convergence in the plastic strain field with decreasing element size for simulated times of 25 ms, 50 ms, and 75 ms. Plastic strains localize to the lower notch and evolve some semblance of a plastic hinge.

believe this idea to be immensely practical given the available resources for system analysts.

The idealization for this concept is noted in Figure 6-7. We have employed this geometry in prior work on remeshing as it yields a nonintuitive load path with the evolution of the deformation. In contrast to a circular bar that preserves symmetry in the necking process, the "puckering" of a rectangular bar is quite interesting and less intuitive. The constitutive parameters for the `fefp` model mirror Section 6.3. Without adaptivity, the solution is illustrated in Figure 6-8. We elect to illustrate the model system (with symmetry) to access the centerline of the bar. The evolution of the deformation proceeds from (a) to (h). Because of the tremendous hardening in the constitutive model, the cross-section remains nearly a square until the latter part of the deformation. Necking initiates on the perturbed, lower cross-section over the first few elements along the axis of the deformation. A classic

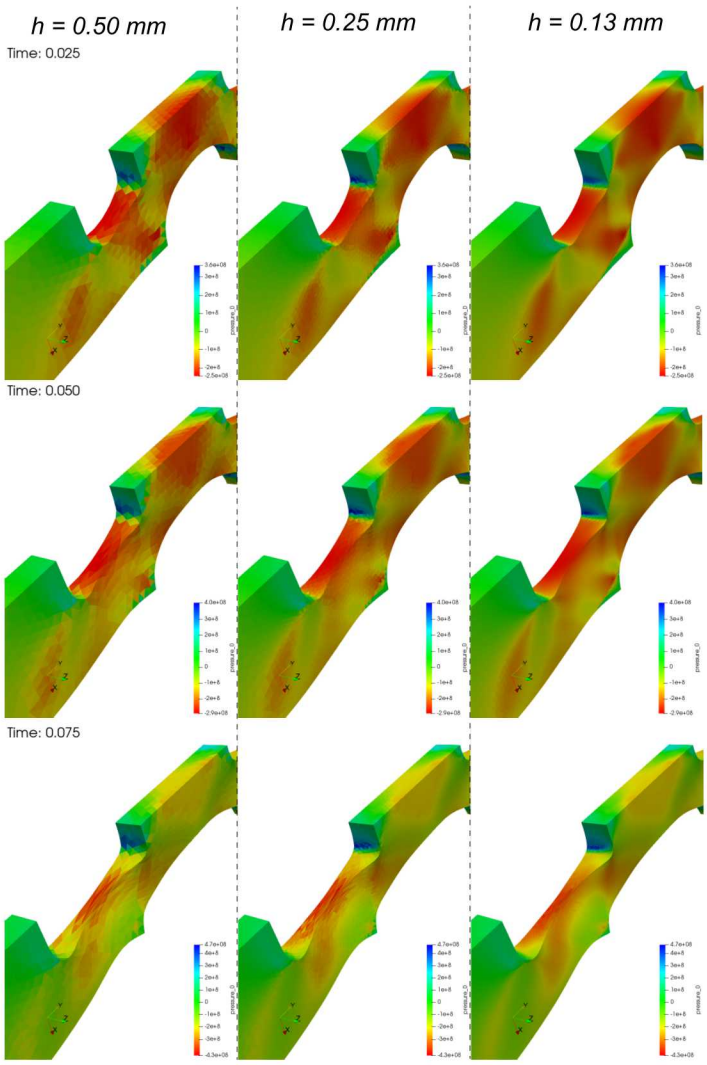


Figure 6-6. The corresponding pressure field under mesh refinement for simulated times of 25 ms, 50 ms, and 75 ms. We note that the displayed quantity is constant within each element and converges with mesh refinement. We can conclude from these findings that that composite tetrahedral element does not suffer from volumetric locking.

tension-compression couple, characteristic of the necking process, forms. We remind the reader of the prior definition of pressure $p = -\text{tr}[\boldsymbol{\sigma}]/3$ with blue indicating hydrostatic tension.

Albeit interesting, localization is essentially relegated to an element band as the critical time step decreases from $6e-7$ s to $6e-17$. The localization process reduces the time step by ten orders of magnitude. For all intents and purposes, the time step becomes zero. Reflecting on system-level simulations, the only mechanisms to maintain some fraction of the initial time step would be to exponentially add mass or remove bands of elements from the simulation.

In contrast, we can employ mesh adaptivity to both resolve the evolving deformation process

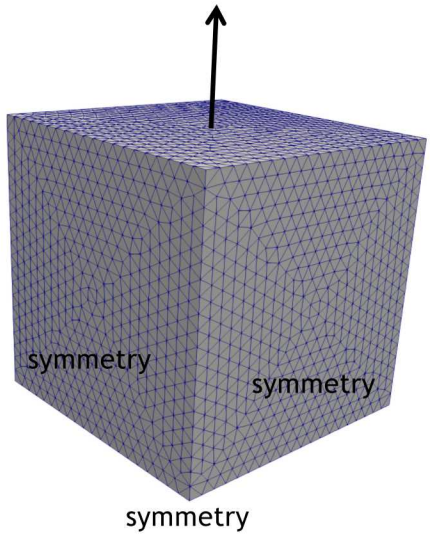


Figure 6-7. Cube with three planes of symmetry subjected to uniaxial tension. The unit cube contains a small geometric perturbation of 0.001 to manage the plane of localization.

and maintain the time step. The evolution the deformation is illustrated in Figure 6-9. Again, we elect to illustrate the model system to view the centerline of the rectangular bar. We focus on the latter part of the deformation in snapshots (a) - (h). Even in regions of the simulation in which the deformation is homogeneous, the mesh is continuously adapted through recourse to Section 5.3 which details the mean ratio element quality measure with a constant size field. The resulting time step is relatively constant and only decreases by roughly a factor of two during the entirety of the simulation from $4e-7$ s to $1.8e-7$ s. This is quite remarkable given the plastic strains exceed 300%. In fact, kinematic fields are typically plotted to illustrate the character of the localization. Instead, we again focus on the pressure field in Figure 6-9. The field is quite smooth and as the deformation becomes lobed and a "butterfly" cross-section emerges. We note that a finer mesh size would better resolve the character of the curvature. That said, the purpose of this study was to determine the deformation path given finite resources and the evolved cross-section is promising given the small reduction in the time step.

Figure 6-10 attempts to illustrate both the simulation time and the run time for both a fixed mesh and adaptivity. For the fixed mesh, the time step craters and the run time exponentially increases for small increments in the simulation time. Consequently, we normalize simulation time by the asymptote observed in the fixed mesh. The run time can then be normalized by the intersection of the asymptote in the fixed simulation with the simulation employing adaptivity. In this sense, Figure 6-10 reveals the cost of obtaining a particular simulation time. As noted in the in Figure 6-10, adaptivity is slightly more expensive in the initial part of the simulation. However, after ~ 0.5 of normalized simulation time, mesh adaptivity is more efficient. In fact, looking at the entirety of the normalized simulation time, we can conclude that mesh adaptivity ensures that run time (read cost) is

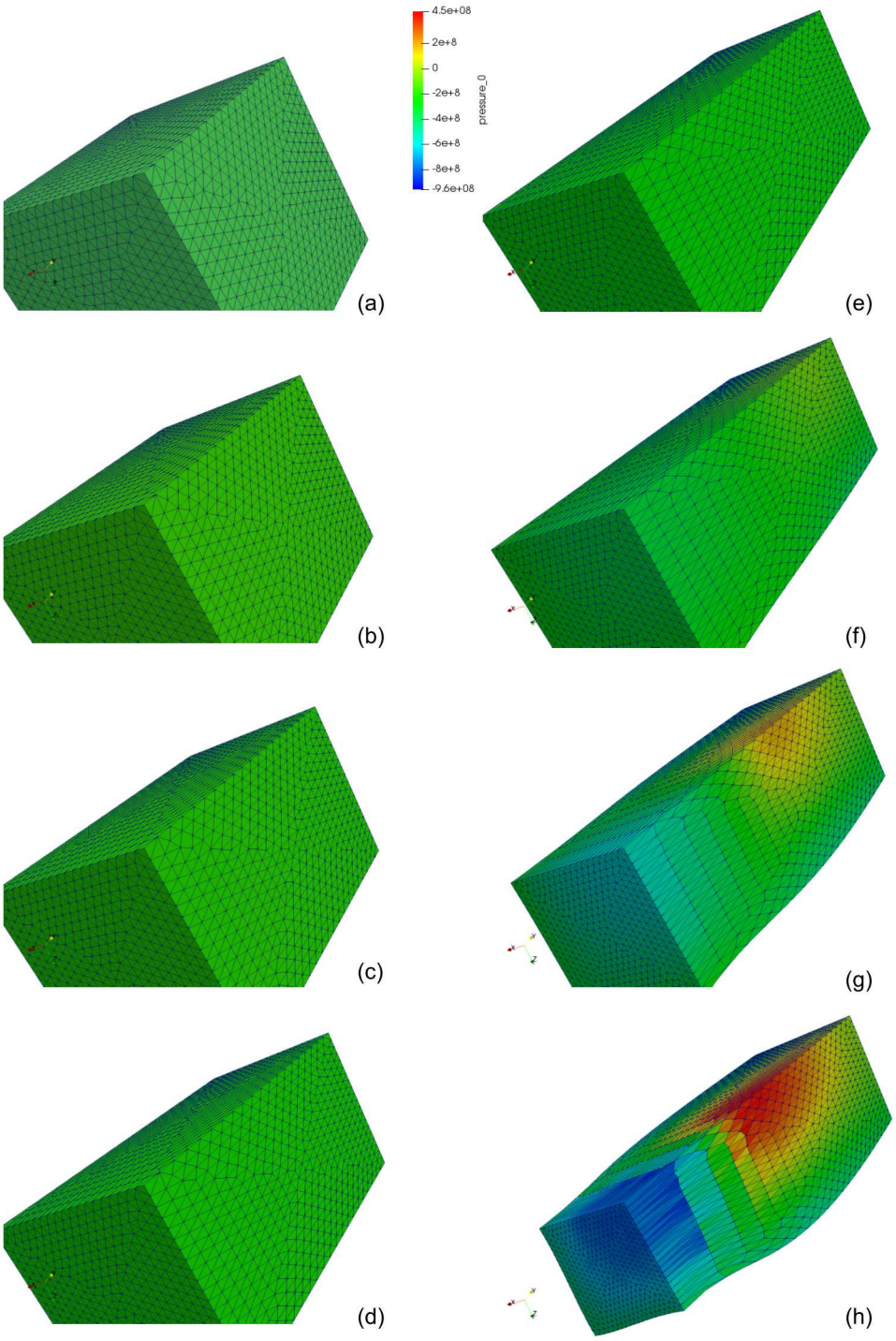


Figure 6-8. Cube geometry subjected to uniaxial tension for a single, initial discretization. Although the square cross-section does indeed neck under large deformations, the time step rapidly decreases ten orders of magnitude from $6e-7$ s to $6e-17$ s.

roughly proportional to the simulation time. Given run time resources, we can achieve proportional simulation times with prescribed levels of discretization to resolve complex

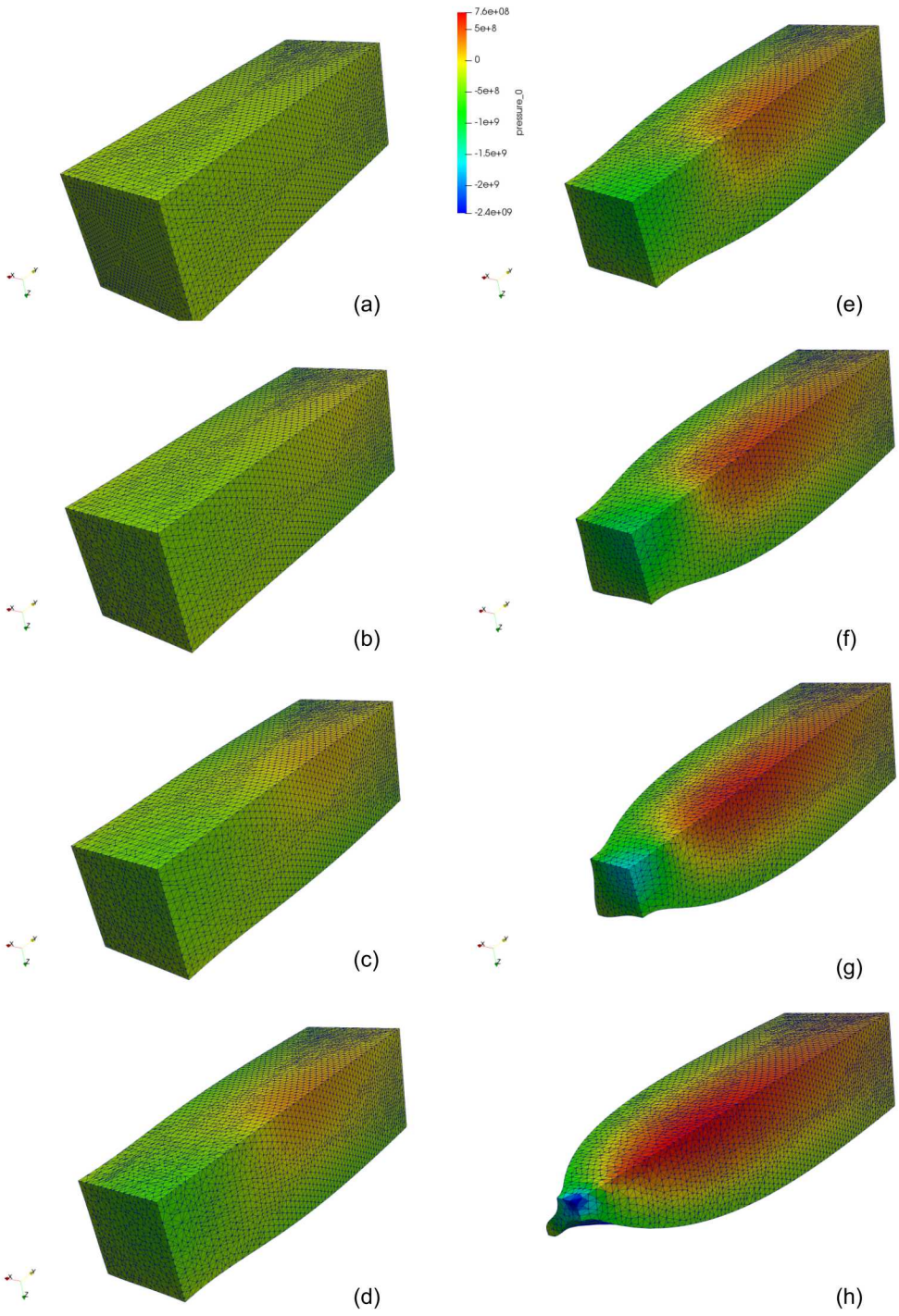


Figure 6-9. Cube geometry subjected to uniaxial tension. Mesh adaptivity triggered by mean ratio element quality measure. Size field is kept constant. Time step decreases by roughly a factor of two from $4e-7$ s to $1.8-e7$ s.

phenomena.

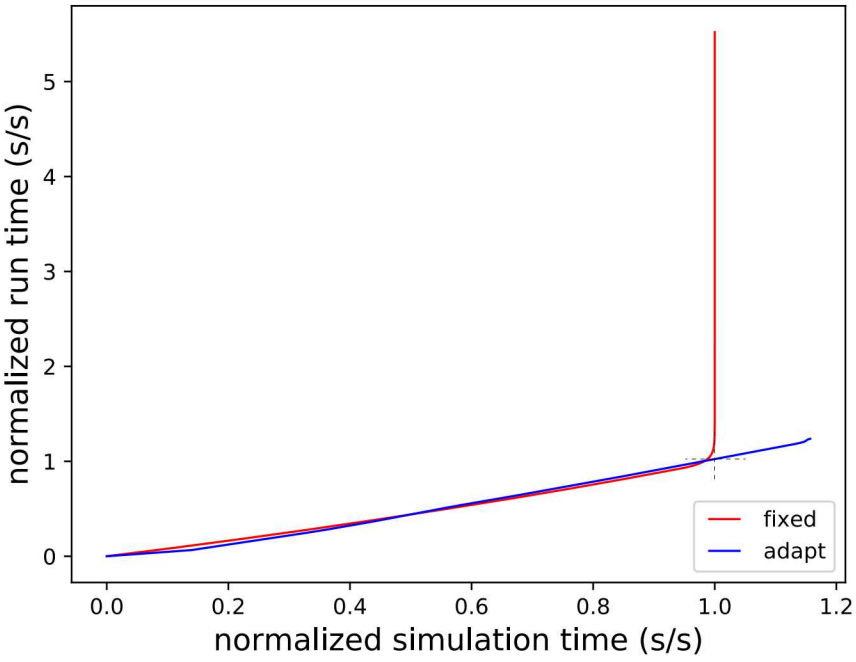


Figure 6-10. Comparing a fixed mesh with mesh adaptivity for the necking of a rectangular bar. The simulation time is normalized by the asymptote in the fixed simulation. The run time is normalized by roughly the intersection of the asymptote of the fixed simulation with the adaptive simulation. Adaptivity ensures that simulation time is roughly proportional to the run time.

6.5. ADAPTIVE REFINEMENT TO CAPTURE LOCALIZATION

We now revisit the necking of the notched bar illustrated for a set of fixed discretizations in Section 6.3. For this thought experiment, we seek to resolve the localization of the deformation. Again, we attempt to align this idealized example with a system-level model. Let us hypothesize that we have a large system-level model with tens of millions of degrees of freedom. Although we can admit rather small mesh sizes, we cannot accept uniform refinement. In three dimensions, we quickly move from the attainable to the unattainable. In addition, because the system-level model is exposed to multiple scenarios under varying environments, the region of localization will move in the body. A practical path forward is to invoke adaptivity and resolve the localization process in-core and under deformation.

The size field we employed for this study was outlined in Section 5.4.2. The initial element size, h_{max}, is 0.25 mm. The minimum element size h_{min} is 0.10 mm. The size field decreases linearly with increasing plastic strain until $\epsilon_{p,max} = 1.0$. We continue to adapt the mesh for $\epsilon_p > 1.0$ with a constant size field. As before, adaptivity is triggered when the mean ratio element quality measure $\eta_{k,min} < 0.3$. We hypothesize that the equivalent plastic strain may be helpful for adaptivity. The plastic strain represents a monotonically increasing state

variable that is not susceptible to oscillatory behavior (read stress waves) in dynamic environments. Our hypothesis is confirmed in Figure 6-11. Each time the element quality is degraded beyond our threshold, we adapt the discretization with a hetrogeneous size field driven by the localization process. That adaption is evident in the history of the deformation. Not only were we able to capture the in-plane plastic hinge but also the through-thickness "puckering" in the root of the notch.

Because mesh modifications are local and confined to the region of localization, the number of additional nodes is minimal. The initial discretization contained 211,906 nodes at a characteristic element size of 0.025 mm. The final configuration noted in Figure 6-11 contains 299,687 nodes with an h_{min} of 0.010 mm. We contrast the final, adaptive discretization with uniform refinement for the region of interest with an element size of 0.010 mm. Uniform refinement yielded 2,118,990 nodes. Thus, uniform refinement had a $\sim 7X$ increase in the number of nodes and a $\sim 21X$ increase in the number of degrees of freedom (DOF). This concept is displayed graphically in Figure 6-12. Mesh adaptivity has the potential to yield solutions that are more robust, more accurate, and more efficient.

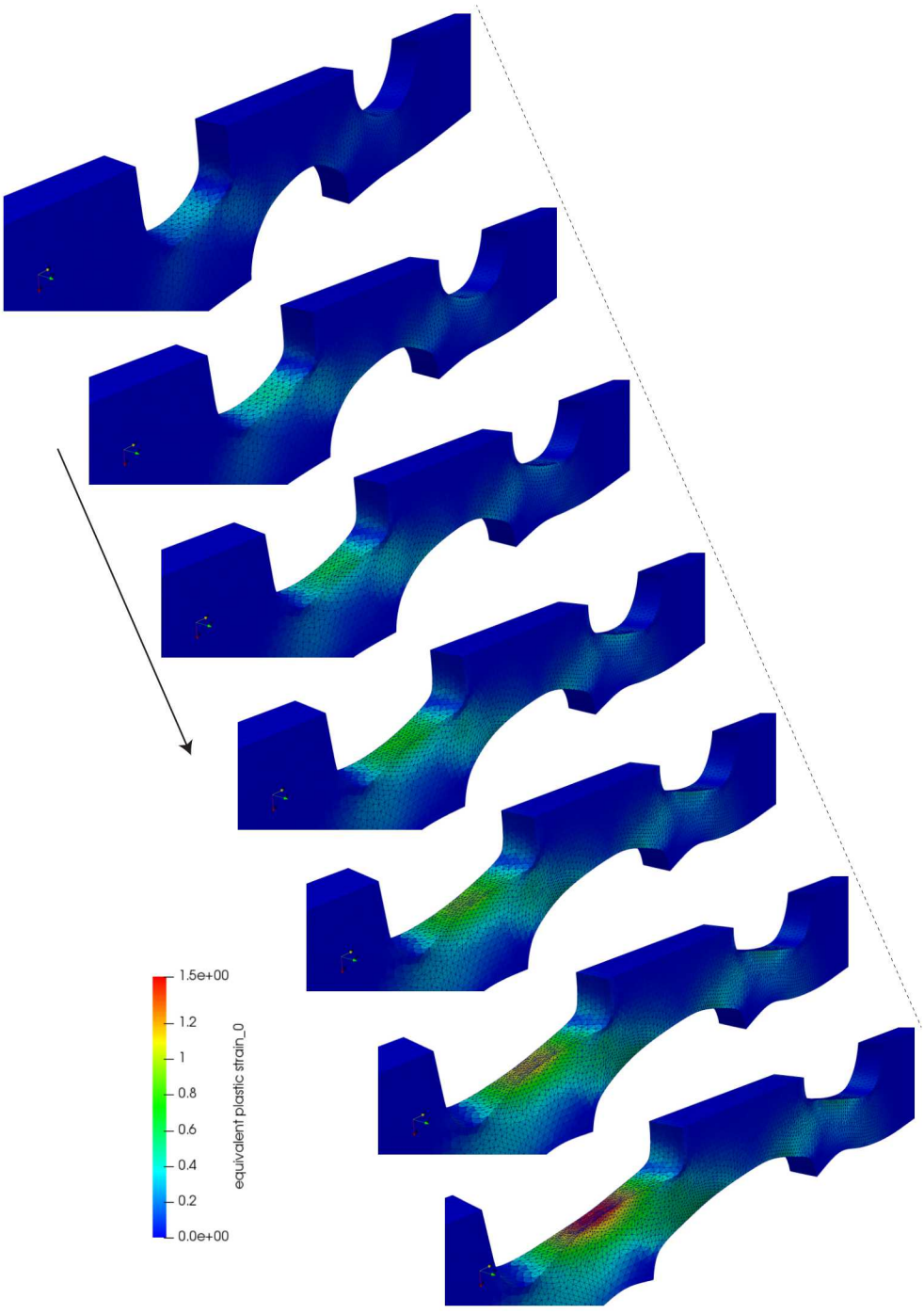


Figure 6-11. To capture both in-plane and through-thickness necking, adaptivity is employed through a size field governed by an internal state variable, the equivalent plastic strain. Not only does adaptivity capture the plastic hinge, the new approach also resolves surface "puckering" in the root of the notch.

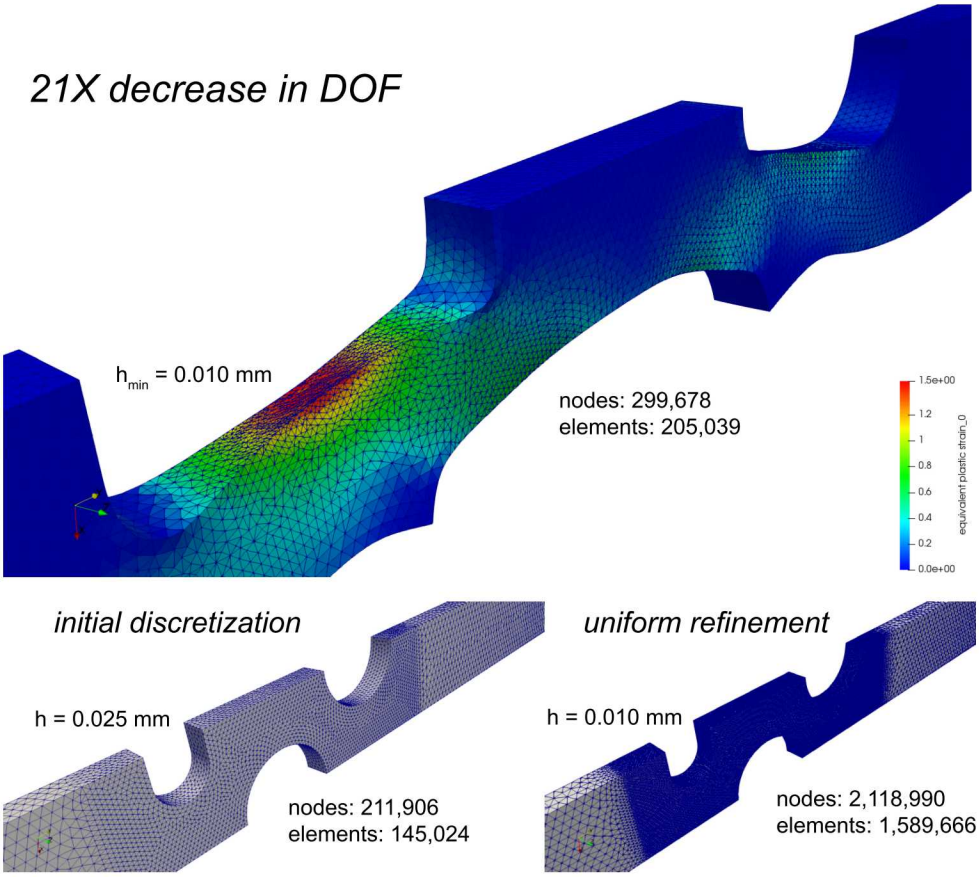


Figure 6-12. If we employ mesh adaptivity in lieu of uniform refinement, we can not only continue to adapt with ongoing resolution beyond $\epsilon_{p,max}$, but also benefit from a 21X reduction in the number of degrees of freedom.

7. FUTURE WORK AND CONCLUSIONS

This effort stemmed from the Director's Challenge. We do believe that in-core, physics specific adaptivity is the key to having a single model adapt to multiple requirements. We also believe that the cases presented in Section 6 are quite applicable to system-level modeling approaches. System-level analysts will always have limited resources. Our goal is to embrace those limitations with adaptivity. Reflecting on FY19, we offer what the 150K investment purchased

1. A broad, multidisciplinary team
2. Potentially unique, adaptive capability in the entire solid mechanics community
3. Composite tetrahedral element suitable for large plastic deformations
4. In-core adaption of higher-order tet with OMEGA_H and mapping of internal variables
5. Fundamental advances in adaptvity needed for the systematic modeling of full engineering models
6. Studies which illustrate gaps in mapping and extensions in OMEGA_H
7. A baseline framework for adaption to improve indicators and develop error estimators.

As previously stated, we hypothesized a research opportunity for mesh adaption to achieve a 10X reduction in the number of degrees of freedom for metallic bodies undergoing finite, elastoplastic deformations in dynamic environments. We proved that hypothesis and successfully demonstrated a 21X reduction with increased robustness and resolution.

Future work seeks to not only improve mapping and extension for higher-order tetrahedral elements in OMEGA_H but also to move beyond isotropic error indicators to anisotropic error indicators to drive anisotropic adaption. We hypothesize that even greater reductions in the degrees of freedom and a path forward to realizing our vision.

BIBLIOGRAPHY

[1] Frédéric Alauzet et al. "Parallel anisotropic 3D mesh adaptation by mesh modification". In: *Engineering with Computers* 21.3 (2006), pp. 247–258.

[2] S. Antman. *Nonlinear Problems in Elasticity*. Second. New York: Springer, 2005.

[3] V Balopoulos and JF Abel. "Use of shallow class hierarchies to facilitate object-oriented nonlinear structural simulations". English. In: *FINITE ELEMENTS IN ANALYSIS AND DESIGN* 38.11 (Sept. 2002), 1047–1074. ISSN: 0168-874X.

[4] Ewen Biotteau et al. "Three dimensional automatic refinement method for transient small strain elastoplastic finite element computations". In: *Computational Mechanics* 49.1 (2012), pp. 123–136.

[5] Ramzy Boussetta, Thierry Coupez, and Lionel Fourment. "Adaptive remeshing based on a posteriori error estimation for forging simulation". In: *Computer methods in applied mechanics and engineering* 195.48-49 (2006), pp. 6626–6645.

[6] A. Bucher et al. "A comparison of mapping algorithms for hierarchical adaptive FEM in finite elasto-plasticity". English. In: *Computational Mechanics* 39.4 (2007), 521–536. ISSN: 0178-7675. DOI: {10.1007/s00466-006-0051-z}.

[7] J-P Combe, Pierre Ladevéze, and J-P Pelle. "Discretization error estimator for transcient dynamic simulations". In: *Advances in Engineering Software* 33.7-10 (2002), pp. 553–563.

[8] Gaëtan Compere et al. "A mesh adaptation framework for dealing with large deforming meshes". In: *International journal for numerical methods in engineering* 82.7 (2010), pp. 843–867.

[9] Pedro Díez, Marino Arroyo Balaguer, and Antonio Huerta. "Adaptivity based on error estimation for viscoplastic softening materials". In: *Mechanics of Cohesive-Frictional Materials* 5.2 (2000), pp. 87–112.

[10] Rémi Feuillet et al. "Connectivity-change moving mesh methods for high-order meshes: Toward closed advancing-layer high-order boundary layer mesh generation". In: *2018 Fluid Dynamics Conference*. 2018, p. 4167.

[11] James W Foulk III et al. "A 10-node composite tetrahedral finite element for solid mechanics". In: *preprint* (2019).

[12] Pascal-Jean Frey and Frédéric Alauzet. "Anisotropic mesh adaptation for CFD computations". In: *Computer methods in applied mechanics and engineering* 194.48-49 (2005), pp. 5068–5082.

[13] J. Gallier. *Geometric Methods and Applications: For Computer Science and Engineering*. Springer, June 2011. ISBN: 9781441999603.

[14] Brian N Granzow, Assad A Oberai, and Mark S Shephard. "Adjoint-based error estimation and mesh adaptation for stabilized finite deformation elasticity". In: *Computer Methods in Applied Mechanics and Engineering* 337 (2018), pp. 263–280.

[15] Brian N Granzow, Mark S Shephard, and Assad A Oberai. "Output-based error estimation and mesh adaptation for variational multiscale methods". In: *Computer Methods in Applied Mechanics and Engineering* 322 (2017), pp. 441–459.

[16] N. J. Higham. "Evaluating Padé approximants of the matrix logarithm". English. In: *SIAM Journal on Matrix Analysis and Applications* 22.4 (2001), 1126–1135. ISSN: 0895-4798.

[17] N. J. Higham. "The scaling and squaring method for the matrix exponential revisited". English. In: *SIAM Journal on Matrix Analysis and Applications* 26.4 (2005), 1179–1193. ISSN: 0895-4798. DOI: {10.1137/04061101X}.

[18] Daniel Alejandro Ibanez. "Conformal Mesh Adaptation on Heterogeneous Supercomputers". PhD thesis. Rensselaer Polytechnic Institute, 2016.

[19] X. M. Jiao and M. T. Heath. "Common-refinement-based data transfer between non-matching meshes in multiphysics simulations". English. In: *International Journal for Numerical Methods in Engineering* 61.14 (2004), 2402–2427. ISSN: 0029-5981. DOI: {10.1002/nme.1147}.

[20] Y. Kosmann-Schwarzbach. *Groups and Symmetries: From Finite Groups to Lie Groups*. Springer, Aug. 2009, p. 207. ISBN: 9780387788654.

[21] P Ladevèze and JP Pelle. "Estimation of discretization errors in dynamics". In: *Computers & structures* 81.12 (2003), pp. 1133–1148.

[22] Sudeep K Lahiri, Javier Bonet, and Jaume Peraire. "A variationally consistent mesh adaptation method for triangular elements in explicit Lagrangian dynamics". In: *International journal for numerical methods in engineering* 82.9 (2010), pp. 1073–1113.

[23] Fredrik Larsson, Peter Hansbo, and Kenneth Runesson. "Strategies for computing goal-oriented a posteriori error measures in non-linear elasticity". In: *Int. J. for Numerical Methods in Eng.* 55.8 (2002), pp. 879–894. DOI: 10.1002/nme.513.

[24] Xiangrong Li, Mark S Shephard, and Mark W Beall. "3D anisotropic mesh adaptation by mesh modification". In: *Computer methods in applied mechanics and engineering* 194.48-49 (2005), pp. 4915–4950.

[25] Adrien Loseille, Victorien Menier, and Frédéric Alauzet. "Parallel generation of large-size adapted meshes". In: *Procedia Engineering* 124 (2015), pp. 57–69.

[26] Qiukai Lu. "Developments of Parallel Curved Meshing for High-Order Finite Element Simulations". PhD thesis. Rensselaer Polytechnic Institute, 2011.

[27] J. E. Marsden and T. S. Ratiu. *Introduction to Mechanics and Symmetry: a Basic Exposition of Classical Mechanical Systems*. Springer, 1999, p. 610. ISBN: 9780387986432.

[28] Arif Masud and Timothy J Truster. "A framework for residual-based stabilization of incompressible finite elasticity: Stabilized formulations and F methods for linear triangles and tetrahedra". In: *Computer Methods in Applied Mechanics and Engineering* 267 (2013), pp. 359–399.

[29] J. Mosler and Michael Ortiz. "Variational h-adaption in finite deformation elasticity and plasticity". In: *International Journal for Numerical Methods in Engineering* 72.5 (2007), pp. 505–523.

[30] A. Mota et al. "Lie-group interpolation and variational recovery for internal variables". In: *Computational Mechanics* 52.6 (Dec. 2013), pp. 1281–1299. ISSN: 0178-7675.

[31] W. Muschik, C. Papenfuss, and H. Ehrentraut. "A sketch of continuum thermodynamics". English. In: *Journal of Non-Newtonian Fluid Mechanics* 96.1-2, Sp. Iss. SI (2001). 2nd International Workshop on Nonequilibrium Thermodynamics, Oxford, England, Aug, 2000, 255–290. ISSN: 0377-0257.

[32] William F Noh. "Errors for calculations of strong shocks using an artificial viscosity and an artificial heat flux". In: *Journal of Computational Physics* 72.1 (1987), pp. 78–120.

[33] A. Orlando and D. Peric. "Analysis of transfer procedures in elastoplasticity based on the error in the constitutive equations: Theory and numerical illustration". English. In: *International Journal for Numerical Methods in Engineering* 60.9 (2004), 1595–1631. ISSN: 0029-5981.

[34] M. Ortiz and J. J. Quigley. "Adaptive Mesh Refinement in Strain Localization Problems". English. In: *Computer Methods in Applied Mechanics and Engineering* 90.1-3 (1991), 781–804. ISSN: 0045-7825.

[35] Jakob T Ostien et al. "A 10-node composite tetrahedral finite element for solid mechanics". In: *International Journal for Numerical Methods in Engineering* 107.13 (2016), pp. 1145–1170.

[36] D. Peric et al. "Transfer operators for evolving meshes in small strain elasto-placticity". English. In: *Computer Methods in Applied Mechanics and Engineering* 137.3-4 (1996), 331–344. ISSN: 0045-7825.

[37] R. Radovitzky and M. Ortiz. "Error estimation and adaptive meshing in strongly nonlinear dynamic problems". English. In: *Computer Methods in Applied Mechanics and Engineering* 172.1-4 (1999), 203–240. ISSN: 0045-7825.

[38] Raul Radovitzky and Michael Ortiz. "Error estimation and adaptive meshing in strongly nonlinear dynamic problems". In: *Computer Methods in Applied Mechanics and Engineering* 172.1-4 (1999), pp. 203–240.

[39] Rolf Rannacher and F-T Suttmeier. "A feed-back approach to error control in finite element methods: Application to linear elasticity". In: *Comput. Mechanics* 19.5 (1997), pp. 434–446. DOI: 10.1007/s004660050191.

[40] M. M. Rashid. "Material state remapping in computational solid mechanics". English. In: *International Journal for Numerical Methods in Engineering* 55.4 (2002), 431–450. ISSN: 0029-5981. DOI: {10.1002/nme.508}.

[41] Antonio Rodríguez-Ferran and Antonio Huerta. "Error estimation and adaptivity for nonlocal damage models". In: *International Journal of Solids and Structures* 37.48-50 (2000), pp. 7501–7528.

[42] M. R. Sepanski. *Compact Lie Groups*. Springer, 2007, p. 208. ISBN: 9780387302638.

[43] SIERRA Solid Mechanics Team. *Sierra/SolidMechanics 4.52 User's Guide*. Technical report (UUR) SAND2019-2715. Albuquerque, New Mexico 87185 and Livermore, California 94550: Sandia National Laboratories, 2019.

[44] RA Spurrier. "Comment on Singularity-free Extraction of a Quaternion from a Direction-cosine Matrix". English. In: *Journal of Spacecraft and Rockets* 15.4 (1978), 255. ISSN: 0022-4650.

[45] Francesc Verdugo, Núria Parés, and Pedro Díez. "Error assessment in structural transient dynamics". In: *Archives of computational methods in engineering* 21.1 (2014), pp. 59–90.

[46] JP Whiteley and SJ Tavener. "Error estimation and adaptivity for incompressible hyperelasticity". In: *Int. J. for Numerical Methods. in Eng.* 99.5 (2014), pp. 313–332. DOI: 10.1002/nme.4677.

DISTRIBUTION

Hardcopy—External

Number of Copies	Name(s)	Company Name and Company Mailing Address

Hardcopy—Internal

Number of Copies	Name	Org.	Mailstop

Email—Internal (encrypt for OUO)

Name	Org.	Sandia Email Address
Technical Library	01177	libref@sandia.gov

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.