INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING
Int. J. Numer. Meth. Engng 2008; **00**:1–39 *Prepared using* nmeauth.cls *[Version: 2002/09/18 v2.02]*

A Variationally Consistent Mesh Adaptation Method for Triangular Elements in Explicit Lagrangian Dynamics

Sudeep K. Lahiri [1], Javier Bonet [2,*] and Jaime Peraire [1]

[1] *Aerospace Computational Design Laboratory, Department of Aeronautics & Astronautics, MIT, Cambridge, USA.*
[2] *Civil & Computational Engineering Center, School of Engineering, Swansea University, UK.*

SUMMARY

In this paper a variational formulation for mesh adaptation procedures, involving local mesh changes for triangular meshes, is presented. Such local adaptive changes are very well suited for explicit methods as they do not involve significant computational expense. They also greatly simplify the projection of field variables from the old to the new meshes. Crucially, the variational nature of the formulation used to derive the equilibrium equations at steps where adaptation takes place ensures that conservation of linear and angular momentum is obtained [1]. Several examples in 2-D showing the application of the proposed adaptive algorithms are used to demonstrate the validity of the methodology proposed.

KEY WORDS: Variationally consistent, mesh adaptation, explicit, space-time discretization, triangular elements

1. Introduction

Rapid dynamics encompasses a significant section of continuum mechanics problems. Several industrial phenomena involve rapid dynamics of solids, for example forging, machining, crash-tests, collision modeling and many others. Computational simulations of such problems are used in various engineering analysis and design. These problems involve large deformations and rotations along with complex material behavior. Hence these problems are inherently non-linear. Due to high velocities (of the order of speed of sound in the material), large meshes and many small time-steps are used for spatial and temporal accuracy. Hence explicit time-integrators become advantageous in such applications. Several codes have been developed and used for such problems [2, 3, 4, 5], based on explicit methods. The main challenges in these

[*] Correspondence to: Javier Bonet, Civil & Computational Engineering Center, School of Engineering, Swansea University SA2 8PP, UK.(j.bonet@swansea.ac.uk)

Contract/grant sponsor: Sandia National Laboratory, USA.; contract/grant number: Doc. No. 1152 under A0260

numerical problems lie in the proper modeling of large deformations and rotations, contact, and complex non-linear material behavior. Mesh distortions, encountered due to large deformations lead to lack of accuracy of the solution. Mesh adaptive time integration can be used to reduce mesh distortions and increase the accuracy of the solution. Such use of mesh adaptation has been limited, since these updates add errors to the solution. Existing mesh-adaptive methods combined with common time integration techniques do not ensure conservation of momentum which may lead to errors over many time-integration steps. Hence, it is desired that such mesh adaptation methods conserve global momentum which would allow use of adaptation in reducing mesh distortions and also increase the accuracy of the solution.

An important aspect of a time-integration method in dynamics applications is its ability to conserve mass, momentum (linear and angular) and energy, which leads to more physically consistent solutions. Methods which do not have good conservation properties, develop large errors over many time integration steps. Typically, dynamics in solids are modeled from a Lagrangian formulation of the equations of motion. Hence mass conservation is automatically satisfied in such methods. Exact conservation of global energy is hard to obtain using explicit integrators. But global momentum (linear and angular) conservation is possible. The explicit time-integrator, the Central Difference Scheme (also called the Leap-Frog Method), is found to conserve global momentum exactly. Existing codes [4, 5] have employed this method with great success.

Recent research [1, 6] has shown that time-integration methods developed from a variational principle as that of Hamilton's principle of stationary action, necessarily conserve linear and angular momentum. Such methods are commonly called as **Variational Integrators** or **Variational methods**. In this paper, topological changes for mesh adaptation are developed from Hamilton's principle and space-time discretization, leading to **Variational Mesh Adaptation** which conserves the total momentum (linear and angular) of the discrete system.

1.1. Literature review

1.1.1. Variational Framework Variational integrators have been developed by several researchers [1, 7, 8, 9, 10, 11, 12, 13], on the basis of Hamilton's principle of stationary action, rather than discretizing the differential equations of motion in time. Hamilton's principle dictates that the path followed by a body represents a stationary point of the action integral of the Lagrangian over a given time interval [14, 15]. Variational integrators take advantage of this principle by constructing a discrete approximation of this integral which then becomes a function of a finite number of positions of the body at each time step. The stationary condition of the resulting discrete functional with respect to each body configuration leads to time stepping algorithms that retain many of the conservation properties of the continuum problem. In particular, the schemes developed in this way satisfy exact conservation of linear and angular momentum [1]. In addition, these algorithms are found to have excellent energy conservation properties even though the exact reasons for this are not fully understood [1, 16, 17, 18]. This class of variational algorithms includes both implicit and explicit schemes, and in particular, it includes some well-known members of the Newmark family [19]. A recent development in the area of variational integrators, is the development of asynchronous variational integrators [10]. The discrete energy gets computed as the variation of the Lagrangian with respect to the time-step. By altering the time step locally, it has been shown in [10], that variational integrators could have both momentum and energy conserving properties but at the cost of being asynchronous. This paper discusses only synchronous time-step methods.

1.1.2. Mesh Adaptation Mesh adaptation has been an active area of research in solid and fluid mechanics computations. There are three types of mesh adaptation *viz.* : (1) r-adaptation, where the number of nodes and number of elements remain same while the node locations or connectivities are changed [20], (2) h-adaptation, where the elements are refined and de-refined locally or globally [21], and (3) p-adaptation, where the order of the interpolation polynomial within the element is changed to resolve the solution locally [22]. The effectiveness of mesh adaptation depends on the mesh-adaptive-mechanism, and the adaptation criteria.

Mesh-adaptive mechanisms might include local mesh changes or global remeshing. Global mesh changes, typically involve, complete remeshing and transfer of variables from the old mesh to the new mesh [23, 24]. Local mesh changes could be achieved using explicit updates [25, 26]. Mesh changes involve node movement, changes in mesh connectivity, and coarsening and refinement of meshes. A detailed overview of such changes in meshes can be found in [27, 28]. Various such mesh update methods exist, which are used by several researchers [4, 29, 30] with success. 2D remeshing based on the advancing front methods have been used in [30] for modeling ballistic penetration problems. Severe mesh distortions encountered in 2D machining problems have been handled in [31], based on complete remeshing techniques. 2D mesh adaptation for shear bands in plane strain can be found in [32, 33] Local coarsening and refinement based on mesh size has been discussed in [29] in application to shear bands. 3D Mesh operations are discussed in [29, 27]. Mesh adaptations for metal forming can be found in [34, 35, 36]. 2D Impact problems have been modeled using global remeshing and gradient based indictors in [37]. Mesh adaptation has also been used in shape optimization of structures [38, 39].

The adaptation criteria is chosen by the analyst. Typically meshes are adapted based on either some error-estimate or mesh skewness or some output of interest. Various researchers [40, 41, 42, 43, 44, 23, 45] have described different error estimation techniques in their works. A commonly used error estimate by Zienkiewicz and Zhu, [46, 47], (Z^2 error estimate), uses the stresses within the element and describes a recovery process to obtain a reference stress. More recent developments, [48, 49, 50] have used a new approach for error-estimation based on the constitutive relation error. Error estimates based on variational constitutive updates can be found in [23]. Variational mesh adaptation, where the error-estimate is obtained from a variational principle is found in [23, 51, 52, 53]. Recently, some researchers [39, 54], have used the idea of configurational forces [55] for r-adaptation, for applications in shape optimization. An overview of various error-estimation techniques and adaptation criteria can be found in [45, 56].

1.2. Overview

Section 2 of this paper reviews the variational framework from time step integrators. The details of the space-time discretization used later in our adaptive formulation are then presented. The derivation of the simple leap frog method using space-time discretization is shown as an example. In section 3, the space-time discretization and the variational formulation are extended to incorporate local mesh adaptations. Local remeshing is achieved by four local operations, *viz.*: (1) Diagonal Swapping, (2) Edge Splitting, (3) Node Movement and (4) Edge Collapsing. Details of the above mechanisms are presented individually. Then, implementation details of error-estimation, and adaptation criteria are mentioned followed by examples demonstrating the performance of the adaptation methods. In section 6, A brief summary of the overall developments of the research is presented, followed by suggestions of possible future work.

2. Variational Formulation

2.1. The Continuous Problem

The motion under loading of a generic three dimensional body is considered. A reference configuration, $Q \subset \mathcal{R}^3$ is adopted, corresponding to the configuration of the body at time $t = 0$. The material coordinates $\boldsymbol{X} \in Q$, are used to label the particles of the body. At any arbitrary time t, the position of particle $\boldsymbol{X}$ is given by the coordinate $\boldsymbol{x}$, and in general, the motion of the body is described by a deformation mapping,

$$\boldsymbol{x} \quad = \quad \phi(\boldsymbol{X}, t), \tag{1}$$

as illustrated in figure 1. In its reference configuration, the body has volume V_0 and density ρ_0, whereas at a given time t, the body has volume $V(t)$ and density $\rho(t)$.

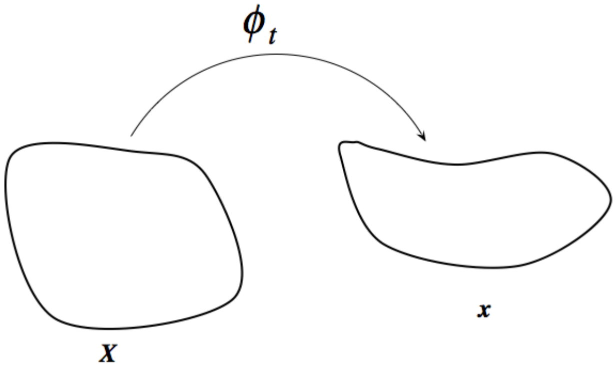

Figure 1. Continuous systems

2.2. The Action Integral for non-dissipative systems

For non-dissipative systems, both the internal and external forces in the system can be derived from a potential, and the motion between times $t_0 = 0$ and t, can be determine from Hamilton's principle. To this end, a Lagrangian, $\mathcal{L}$, is introduced, such that, $\mathcal{L}(\boldsymbol{x}, \dot{\boldsymbol{x}}) = \mathcal{K}(\dot{\boldsymbol{x}}) - \Pi(\boldsymbol{x})$, where, $\mathcal{K}$, denotes the kinetic energy, Π is the potential energy and $\dot{\boldsymbol{x}} = d\boldsymbol{x}/dt$ is the material velocity. The potential energy can be generally decomposed into an internal elastic component, Π^{int}, and a component accounting for the external conservative forces, Π^{ext}. Thus, $\Pi(\boldsymbol{x}) = \Pi^{\text{int}}(\boldsymbol{x}) + \Pi^{\text{ext}}(\boldsymbol{x})$.

The action integral, S, is defined as the integral of the of the Lagrangian over the time interval considered,

$$S \;=\; \int_0^t \mathcal{L}(\boldsymbol{x}, \dot{\boldsymbol{x}})\, dt \;, \tag{2}$$

and Hamilton's principle states that the deformation mapping satisfying the equations of motion can be obtained by making the action integral stationary with respect to all possible deformation mappings which are compatible with the boundary conditions [15], where the Lagrangian $\mathcal{L}$ can be expressed in terms of the deformation and velocities in the following manner.

2.2.1. The Kinetic Energy, $(\mathcal{K})$ The kinetic energy of the body is a function of the material velocity and can be written as:

$$\mathcal{K}(\dot{\boldsymbol{x}}) \;=\; \int_{V_0} \frac{1}{2}\rho_0 \dot{\boldsymbol{x}}^2 dV_0 \;. \tag{3}$$

2.2.2. The Internal Potential Energy (Π^{int}) The internal potential energy depends on the constitutive relations of the materials in the system. In this research hyperelastic compressible Neo-Hookean materials are considered, which undergo large deformations and displacements. Let $\boldsymbol{F}$ be the deformation gradient tensor which can be written as,

$$F_{ij} \;=\; \frac{\partial x_i}{\partial X_j} \qquad \forall\; i, j = 1, .., 3$$

The relevant kinematic quantities associated with the deformation gradient are the right Cauchy-Green tensor, $\boldsymbol{C}$, the Jacobian, J, and the isochoric component of $\boldsymbol{C}$, $\hat{\boldsymbol{C}}$, which are given by,

$$\boldsymbol{C} = \boldsymbol{F}^T \boldsymbol{F}; \qquad J = \det(\boldsymbol{F}); \qquad \hat{\boldsymbol{C}} = J^{-\frac{2}{3}} \boldsymbol{C} \;.$$

For isotropic compressible Neo-Hookean materials, the internal potential energy can be expressed in terms of the Lame constant μ, and the bulk modulus κ as

$$\begin{aligned} \Pi^{\mathrm{int}}(\boldsymbol{x}) \;&=\; \int_{V_0} \pi(\boldsymbol{F}) dV_0 \\ &=\; \int_{V_0} \left[\frac{\mu}{2} \left(tr(\hat{\boldsymbol{C}}) - 3 \right) + \frac{1}{2}\kappa (J-1)^2 \right] dV_0 \;. \end{aligned} \tag{4}$$

The above expression is well suited for compressible or nearly incompressible materials, [57, 58].

2.2.3. The External Potential Energy (Π^{ext}) The external potential energy includes the work done by the external body and surface forces.

$$\Pi^{\mathrm{ext}}(\boldsymbol{x}) \;=\; -\int_{V_0} \boldsymbol{f}^b \cdot \boldsymbol{x}\, dV_0 - \int_{\partial V_0} \boldsymbol{f}^s \cdot \boldsymbol{x}\, dS_0 \tag{5}$$

Here, $\boldsymbol{f}^b$ are the body forces (per unit volume), $\boldsymbol{f}^s$ are the surface forces (per unit surface), and ∂V_0 denotes the section of the boundary, in the reference configuration, where the surface forces are applied.

2.3. Discretization in time

Consider now a sequence of timesteps $t_{n+1} = t_n + \Delta t,\ n = 0, 1, ..., N$, where for simplicity a constant step size has been taken. The position of the body at each step is defined by a mapping $\boldsymbol{x}_n = \phi(\mathbf{X}, t_n)$. A variational algorithm is defined by a discrete sum integral,

$$S(\boldsymbol{x}_0, \boldsymbol{x}_1, ..., \boldsymbol{x}_N) \approx \sum_{n=0}^{N-1} L_{n,n+1}(\boldsymbol{x}_n, \boldsymbol{x}_{n+1}) \tag{6}$$

where the discrete Lagrangian Integral L approximates the integral of the continuum Lagrangian $\mathcal{L}$ over a timestep [1], that is,

$$\begin{aligned} L_{n,n+1}(\boldsymbol{x}_n, \boldsymbol{x}_{n+1}) &\approx \int_{t_n}^{t_{n+1}} \mathcal{L}(\boldsymbol{x}, \dot{\boldsymbol{x}})\, dt \\ &\approx \int_{t_n}^{t_{n+1}} \mathcal{K}(\dot{\boldsymbol{x}})\, dt - \int_{t_n}^{t_{n+1}} \Pi(\boldsymbol{x})\, dt \end{aligned} \tag{7}$$

Here, for simplicity, the case in which the Lagrangian is a function of $\boldsymbol{x}$ and $\dot{\boldsymbol{x}}$ only, is considered. Other cases, like the ones where the Lagrangian is dependant on pressure in addition to the position and velocity are discussed in [6, 59]. The discrete Lagrangian Integral can be further split into the Kinetic Energy Integral and the Potential Energy Integrals as:

$$L_{n,n+1}(\boldsymbol{x}_n, \boldsymbol{x}_{n+1}) = K_{n,n+1}(\boldsymbol{x}_n, \boldsymbol{x}_{n+1}) - \int_{t_n}^{t_{n+1}} \Pi(\boldsymbol{x})\, dt \tag{8}$$

where $K_{n,n+1}$ is an approximation to the Kinetic Energy Integral $\int_{t_n}^{t_{n+1}} \mathcal{K}(\dot{\boldsymbol{x}})\, dt$. There are many ways in which the approximation (8) can be chosen, and, each one will lead to a different time integration algorithm. It has been shown in [6] that the approximation for the Potential Energy Integral :

$$\int_{t_n}^{t_{n+1}} \Pi(\boldsymbol{x})\, dt \approx \Delta t\, \Pi(\boldsymbol{x}_n) \tag{9}$$

where $\Delta t\ = t_{n+1} - t_n$, leads to explicit time marching algorithms with appropriate choice of the discrete Kinetic Energy Integral. Hence with explicit methods in consideration, the discrete Lagrangian within two steps can be rewritten as:

$$L_{n,n+1}(\boldsymbol{x}_n, \boldsymbol{x}_{n+1}) = K_{n,n+1}(\boldsymbol{x}_n, \boldsymbol{x}_{n+1}) - \Delta t\, \Pi(\boldsymbol{x}_n) \tag{10}$$

The stationary conditions of the discrete sum integral S with respect to a variation $\delta \boldsymbol{v}_n$ of the body position at time step n are now given by,

$$D_n S[\delta \boldsymbol{v}_n] = D_2 L_{n-1,n}(\boldsymbol{x}_{n-1}, \boldsymbol{x}_n)[\delta \boldsymbol{v}_n] + D_1 L_{n,n+1}(\boldsymbol{x}_n, \boldsymbol{x}_{n+1})[\delta \boldsymbol{v}_n] = 0 \qquad \forall \delta \boldsymbol{v}_n\ , \tag{11}$$

where D_i denotes directional derivative with respect to i-th variable. The above equation represents the statement of equilibrium at step n and will enable the positions at step $n+1$ to be evaluated in terms of positions at $n-1$ and n. Rewriting the stationary conditions in terms of the Kinetic and Potential Energy Integrals we obtain:

$$D_2 K_{n-1,n}(\boldsymbol{x}_{n-1}, \boldsymbol{x}_n)[\delta \boldsymbol{v}_n] + D_1 K_{n,n+1}(\boldsymbol{x}_n, \boldsymbol{x}_{n+1})[\delta \boldsymbol{v}_n] - \Delta t\, D_1 \Pi(\boldsymbol{x}_n)[\delta \boldsymbol{v}_n] = 0 \qquad \forall \delta \boldsymbol{v}_n \tag{12}$$

2.4. Discretization in time and space

So far in this paper, discretizations in time have been discussed. In this section a simple spatial discretization is introduced using 3 noded triangular elements. Based on triangular elements, the position vector $\boldsymbol{x}_e^n$ in an element e, can be written as:

$$\boldsymbol{x}_n^e \quad = \quad N_a^e \boldsymbol{x}_n^a \tag{13}$$

where N_a^e are linear shape functions within an element e and $\boldsymbol{x}_n^a$ are the nodal position vectors. The action integral as discretized in time in equation 6 now can be rewritten as:

$$\begin{aligned} S &= S(\boldsymbol{x}_n^a; a = 1, \ldots, N^d; n = 1, \ldots, N) \\ &\approx \sum_{n=0}^{N} L_{n,n+1}(\boldsymbol{x}_n^a, \boldsymbol{x}_{n+1}^a;\ a = 1, \ldots, N^d) \end{aligned} \tag{14}$$

where N^d are the number of nodes and N are the number of time steps. The Lagrangian within the time steps n and $n+1$ can be written as:

$$L_{n,n+1}(\boldsymbol{x}_n^a, \boldsymbol{x}_{n+1}^a) \quad = \quad K_{n,n+1}(\boldsymbol{x}_n^a, \boldsymbol{x}_{n+1}^a) - \Delta t \left(\Pi_n^{\text{ext}}(\boldsymbol{x}_n^a) + \Pi_n^{\text{int}}(\boldsymbol{x}_n^a) \right) \tag{15}$$

The stationarity condition then becomes:

$$\frac{\partial S}{\partial \boldsymbol{x}_n^a} \quad = \quad \frac{\partial L_{n,n+1}}{\partial \boldsymbol{x}_n^a} + \frac{\partial L_{n-1,n}}{\partial \boldsymbol{x}_n^a} \quad = \quad 0 \tag{16}$$

which leads to the relations between the derivatives of the Kinetic and Potential Energy integrals as:

$$\frac{\partial K_{n,n+1}}{\partial \boldsymbol{x}_n^a} - \Delta t \frac{\partial \Pi_n^{\text{ext}}}{\partial \boldsymbol{x}_n^a} - \Delta t \frac{\partial \Pi_n^{\text{int}}}{\partial \boldsymbol{x}_n^a} + \frac{\partial K_{n-1,n}}{\partial \boldsymbol{x}_n^a} \quad = \quad 0 \tag{17}$$

Now, each of the derivatives will be calculated separately.

2.5. The Potential Energy Integral

First, the internal Potential Energy and its derivative with respect to $\boldsymbol{x}_n^a$ are calculated. The Potential Energy is a function of $\boldsymbol{x}_a$ at time level n only due to the approximation in Eqn. 9. Hence, for convenience, the time index n is dropped for the rest of this section on Potential Energy Integral. Therefore, $\boldsymbol{x}_a$ implies $\boldsymbol{x}_n^a$ unless mentioned otherwise. In addition, the following index notation is used. Indices e and f are used to denote elements, indices a and b are used to denote nodes, and, indices i, j, k and l are used to denote vector directions in the current (spatial) configuration and I, J, K, L are used to denote the directions of vectors in the reference (material) configuration. Repeated indices imply summation. First, the deformation gradient within the element e, is considered:

$$\boldsymbol{F}^e \ = \ \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{X}} \quad = \quad \boldsymbol{x}_a^e \otimes \frac{\partial N_a^e}{\partial \boldsymbol{X}} \tag{18}$$

where $\boldsymbol{X}$ is the position vector of the reference (material) configuration. Note that since the shape functions are linear in the element the gradients are constant within an element hence the deformation gradient is a constant within the element. Based on the Neo-Hookean model, the internal potential energy (Π^{int}), can be written as:

$$\Pi^{\text{int}}(\boldsymbol{x}) \;=\; \sum_{e} \int_{V_e^0} \pi(\boldsymbol{F}^e)\, dV_e^0 \tag{19}$$

$$\pi(\boldsymbol{F}^e) \;=\; \frac{\mu}{2}\left\{\text{tr}(\hat{\boldsymbol{C}}_e) - 3\right\} + \frac{\kappa}{2}\,(J_e - 1)^2 \tag{20}$$

where

$$J_e \;=\; \det(\boldsymbol{F^e})\,; \quad \boldsymbol{C}_e \;=\; \boldsymbol{F}^{eT}\boldsymbol{F^e}\,; \qquad \boldsymbol{b}_e \;=\; \boldsymbol{F}^e\boldsymbol{F}^{eT}\,; \quad \hat{\boldsymbol{C}}_e \;=\; J_e^{-\frac{2}{3}}\boldsymbol{C}_e\,;$$

Therefore, the derivative of potential energy *wrt.* $\boldsymbol{x}$ can be written as :

$$\frac{\partial\pi(\boldsymbol{F})}{\partial x_i^a} \;=\; \frac{\partial\pi(\boldsymbol{F})}{\partial\boldsymbol{F}} : \frac{\partial\boldsymbol{F}}{\partial x_i^a} \tag{21}$$

$$=\; \boldsymbol{P} : \frac{\partial\boldsymbol{F}}{\partial x_i^a} \tag{22}$$

where $\boldsymbol{P}$ is the first Piola Kirchhoff stress tensor. The first Piola Kirchhoff stresses are related to the Cauchy stress tensor (also called the true stresses) by:

$$\boldsymbol{\sigma} \;=\; J^{-1}\,\boldsymbol{P}\,\boldsymbol{F}^T \tag{23}$$

Further simplifying using indicial notation, leads to:

$$\frac{\partial\pi}{\partial x_i^a} \;=\; P_{iL}\,\frac{\partial N_a^e}{\partial x_j}\,F_{jL}^e \tag{24}$$

Now, introducing a global index of a node as, b, such that it is the a'th node of element e, (represented here as: $(e,a) \in b$) and revisiting equation, 19 & 24, one can express the derivative of the Potential Energy as:

$$\frac{\partial\Pi^{\text{int}}(\boldsymbol{x})}{\partial x_i^b} \;=\; \sum_{(e,a)\in b} \int_{V_e^0} \frac{\partial\pi^e(\boldsymbol{F}^e)}{\partial x_i^a}\, dV_e^0 \tag{25}$$

$$=\; \sum_{(e,a)\in b} \int_{V_e^0} P_{iL}\,\frac{\partial N_a^e}{\partial x_j}\,F_{jL}^e\, dV_e^0 \tag{26}$$

Now, changing the reference volume V^0 to V a current volume one can obtain:

$$\frac{\partial\Pi^{\text{int}}(\boldsymbol{x})}{\partial x_i^b} \;=\; \sum_{(e,a)\in b} \int_{V_e} P_{iL}\,\frac{\partial N_a^e}{\partial x_j}\,F_{jL}^e\,J_e^{-1} dV_e \tag{27}$$

Substituting equation 23 into equation 27, one can obtain:

$$\frac{\partial\Pi^{\text{int}}(\boldsymbol{x})}{\partial x_i^b} \;=\; \sum_{(e,a)\in b} \int_{V_e} \frac{\partial N_a^e}{\partial x_j}\,\sigma_{ij}^e dV_e$$

$$=\; T_i^b \;=\; \sum_{(e,a)\in b} T_{ai}^e \tag{28}$$

where T_i^b are the internal tractions at node b along direction i, and the T_{ai}^e are the elemental internal tractions at a^{th} node of the element along direction i. Similar to the internal Potential Energy, it can be shown [59] that the external Potential Energy (5) would have similar derivatives:

$$\begin{aligned}\frac{\partial \Pi^{\text{ext}}(\boldsymbol{x})}{\partial x_i^b} &= -\sum_{(e,a)\in b} \int_{V_e^0} \rho_0 N_a^e \, f_i^b dV_e^0 - \sum_{(e,a)\in b} \int_{\partial V_e} N_a^e \, f_i^s dS_e \\ &= -F_i^b = -\sum_{(e,a)\in b} F_{ai}^e \end{aligned} \tag{29}$$

where f_i^s are external surface force per unit area, and f_i^b are the body forces per unit volume. Thus the final expression for the derivative of the Potential Energy with respect to the position vector of a global node at time level n ($\boldsymbol{x}_n^b$) is:

$$\frac{\partial \Pi(\boldsymbol{x}_n)}{\partial \boldsymbol{x}_n^b} = \frac{\partial \Pi^{\text{int}}(\boldsymbol{x}_n)}{\partial \boldsymbol{x}_n^b} + \frac{\partial \Pi^{\text{ext}}(\boldsymbol{x}_n)}{\partial \boldsymbol{x}_n^b} = \boldsymbol{T}_n^b - \boldsymbol{F}_n^b \tag{30}$$

2.6. The Kinetic Energy Integral & Space-Time Discretization

In this section a space-time discretization is adopted to formulate the Kinetic Energy Integral described in Eqn. 8. The space-time integrations are conducted over the space-time vector space $V_0 \times R$ (undeformed reference configuration), and not on $V \times R$ (deformed configuration). We begin with a single triangular element with unit thickness.

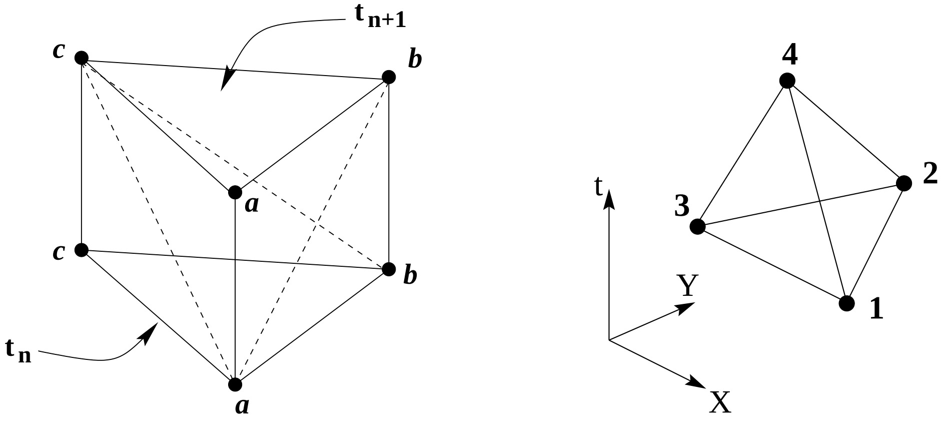

Figure 2. The space-time-prism (left) and a generic space-time-tetrahedron (right).

Figure 2, shows the typical space-time volume of a single triangle. The triangle abc_n and triangle abc_{n+1} enclose a prismatic space-time volume. This volume is further sub-divided into three tetrahedra. The task is to compute the kinetic energy integral K within each of the space-time-tetrahedra, and then sum each of the contributions to compute the net integral within the space-time-prism. To do so, a generic space-time-tetrahedron (Fig. 2, (right)) is studied and the integral is computed by first evaluating the constant velocity over the space-time tetrahedron as:

$$\boldsymbol{x} \quad = \quad \boldsymbol{x}(\boldsymbol{X}, t) \qquad\qquad \boldsymbol{v}_{n,n+1} \quad = \quad \frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}t} \tag{31}$$

Where $\boldsymbol{x}$ is the position vector, $\boldsymbol{X}$ is the reference position vector and $\frac{\mathrm{d}}{\mathrm{d}t}$ is the total derivative. Note here, that for the Kinetic Energy Integral, total derivatives of position vectors, are considered. In the general case, any quantity (scalar or vector) would have a similar treatment. First, a set of volume coordinates are introduced, analogous to the area coordinates in case of triangles. The volume coordinates, given by $(\xi_1, \xi_2, \xi_3, \xi_4)$ attain values of 1 at their corresponding nodes and zero at other nodes, *ie.*, ξ_i is one at node i and zero at all nodes $j \neq i$. Any function *linear* in X, Y, t, say $F(X, Y, t)$, can be interpolated within the tetrahedron, based on its nodal values F_a and shape functions $N_a = \xi_a$ as $F(X, Y, t) = F_a \xi_a$. The coordinate transform between (X, Y, t) and $(\xi_1, \xi_2, \xi_3, \xi_4)$ can be written as:

$$\begin{bmatrix} 1 \\ X \\ Y \\ t \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ X_1 & X_2 & X_3 & X_4 \\ Y_1 & Y_2 & Y_3 & Y_4 \\ t_1 & t_2 & t_3 & t_4 \end{bmatrix} \begin{bmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \\ \xi_4 \end{bmatrix} \tag{32}$$

Inverting this relation gives:

$$\begin{bmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \\ \xi_4 \end{bmatrix} = \frac{1}{6\mathcal{V}_0} \begin{bmatrix} 6\mathcal{V}_1 & a_1 & b_1 & c_1 \\ 6\mathcal{V}_2 & a_2 & b_2 & c_2 \\ 6\mathcal{V}_3 & a_3 & b_3 & c_3 \\ 6\mathcal{V}_4 & a_4 & b_4 & c_4 \end{bmatrix} \begin{bmatrix} 1 \\ X \\ Y \\ t \end{bmatrix} \tag{33}$$

where a_i's are the cofactor of the X_i elements in the transformation matrix. Similarly b_i's are the cofactors of the Y_i elements, c_i's are the cofactors of the t_i elements, and $\mathcal{V}_i$'s are one sixth the cofactor of each unit element in the transformation matrix. $\mathcal{V}_0$ is the volume of the tetrahedron given by:

$$\mathcal{V}_0 \quad = \quad \frac{1}{6} \begin{vmatrix} 1 & 1 & 1 & 1 \\ X_1 & X_2 & X_3 & X_4 \\ Y_1 & Y_2 & Y_3 & Y_4 \\ t_1 & t_2 & t_3 & t_4 \end{vmatrix} \tag{34}$$

Using chain rule, the derivatives of the function now can be written as :

$$\frac{dF}{d\boldsymbol{X}^i} \quad = \quad \frac{\partial F}{\partial \xi^j} \frac{d\xi^j}{d\boldsymbol{X}^i} \tag{35}$$

where $\boldsymbol{X} = [1 \ X \ Y \ t]^T$. Thus, the time gradient of the function F can be written as:

$$\frac{dF}{dt} \quad = \quad \frac{1}{6\mathcal{V}_0} \frac{\partial F}{\partial \xi_j} c_j \tag{36}$$

Now, since F is linearly interpolated, $\left(\frac{\partial F}{\partial \xi_i} = F_i\right)$ which leads to a simple relation for the time derivatives:

$$\frac{dF}{dt} = \frac{\begin{vmatrix} 1 & 1 & 1 & 1 \\ X_1 & X_2 & X_3 & X_4 \\ Y_1 & Y_2 & Y_3 & Y_4 \\ F_1 & F_2 & F_3 & F_4 \end{vmatrix}}{\begin{vmatrix} 1 & 1 & 1 & 1 \\ X_1 & X_2 & X_3 & X_4 \\ Y_1 & Y_2 & Y_3 & Y_4 \\ t_1 & t_2 & t_3 & t_4 \end{vmatrix}} \tag{37}$$

Similarly, assuming a linear interpolation of $\boldsymbol{x}$ $(= F(X,Y,t))$ in space and time, the velocity within the tetrahedron is obtained as a ratio of two determinants:

$$\boldsymbol{v}_{n,n+1} = \frac{d\boldsymbol{x}}{dt} = \frac{\begin{vmatrix} 1 & 1 & 1 & 1 \\ X_1 & X_2 & X_3 & X_4 \\ Y_1 & Y_2 & Y_3 & Y_4 \\ \boldsymbol{x}_1 & \boldsymbol{x}_2 & \boldsymbol{x}_3 & \boldsymbol{x}_4 \end{vmatrix}}{\begin{vmatrix} 1 & 1 & 1 & 1 \\ X_1 & X_2 & X_3 & X_4 \\ Y_1 & Y_2 & Y_3 & Y_4 \\ t_1 & t_2 & t_3 & t_4 \end{vmatrix}} \tag{38}$$

Note here, that in the special case where two nodes of a given space-time-tetrahedron have the same reference coordinate (implying the same point) then, the velocity in the tetrahedron simply becomes (in this case assuming $X_1 = X_4$ and $Y_1 = Y_4$):

$$\boldsymbol{v}_{n,n+1} = \frac{\boldsymbol{x}_4 - \boldsymbol{x}_1}{t_4 - t_1} \tag{39}$$

This simplification leads to a criterion for the choice of subdivision of any generic space-time volume. One should choose to sub-divide a given space-time volume into as many tetrahedra with common nodes as possible. This would lead to a simple velocity interpolation within the tetrahedron. Now the Kinetic Energy Integral K is computed within the space-time-tetrahedron:

$$K^{\text{tet}}_{n,n+1} = \int_{\mathcal{V}_0} \frac{1}{2}\rho_0 \left(\boldsymbol{v}_{n,n+1} \cdot \boldsymbol{v}_{n,n+1}\right) d\mathcal{V}_0 \tag{40}$$

In the case of a tetrahedron with common nodes this volume, simply becomes:

$$\mathcal{V}_0 = \frac{A_{123}}{3}(t_4 - t_1) \tag{41}$$

where A_{123} is the area of the triangle with nodes 1,2 and 3.

$$A_{123} = \frac{1}{2}\begin{vmatrix} 1 & 1 & 1 \\ X_1 & X_2 & X_3 \\ Y_1 & Y_2 & Y_3 \end{vmatrix}$$

In the generic case the Kinetic Energy Integral $K_{n,n+1}$ would take the form:

$$K^{\text{tet}}_{n,n+1} = \mathcal{V}_0 \frac{1}{2}\rho_0 \left(\boldsymbol{v}_{n,n+1} \cdot \boldsymbol{v}_{n,n+1}\right) \tag{42}$$

But in the case of a tetrahedron with common nodes, the Kinetic Energy Integral $K_{n,n+1}$ would take the simple form ($m_{123} = \rho_0 A_{123}$):

$$K^{\text{tet}}_{n,n+1} \quad = \quad (t_4 - t_1)\frac{m_{123}}{3}\frac{1}{2}\left(\boldsymbol{v}_{n,n+1} \cdot \boldsymbol{v}_{n,n+1}\right) \tag{43}$$

Now, revisiting the space-time-prism of the triangle (Fig. 2) it is observed, that it is subdivided into three tetrahedra, each one of them have a common node. Hence, using the above relations, a very simple form of the Kinetic Energy Integral is obtained:

$$K^{\text{prism}}_{n,n+1} \quad = \quad \frac{m_{abc}}{3}\frac{\Delta t}{2}\left[(\boldsymbol{v}^a_{n+1/2} \cdot \boldsymbol{v}^a_{n+1/2}) + (\boldsymbol{v}^b_{n+1/2} \cdot \boldsymbol{v}^b_{n+1/2}) + (\boldsymbol{v}^c_{n+1/2} \cdot \boldsymbol{v}^c_{n+1/2})\right] \tag{44}$$

Where m_{abc} is the mass of the triangle abc and:

$$\begin{aligned} \Delta t \quad &= \quad t_{n+1} - t_n \\ \boldsymbol{v}^{a_i}_{n+1/2} \quad &= \quad \frac{\boldsymbol{x}^{a_i}_{n+1} - \boldsymbol{x}^{a_i}_n}{\Delta t} \qquad\qquad \forall a_i \; = \; a, b, c \end{aligned}$$

Hence the kinetic energy integral for a generic prism for a corresponding triangular element e with mass m^e can be written as:

$$K^e_{n,n+1} \quad = \quad \frac{m^e}{3}\frac{\Delta t}{2}\sum_{a=1:3}\left(\boldsymbol{v}^a_{n+1/2} \cdot \boldsymbol{v}^a_{n+1/2}\right) \tag{45}$$

Note, that the velocities used in each element is simply the nodal value. In case of a finite element mesh, the space-time volume of the entire mesh can be subdivided into space-time-prisms corresponding to each triangular element:

$$K_{n,n+1}(\boldsymbol{x}_n, \boldsymbol{x}_{n+1}) = \sum_e K^e_{n,n+1} \tag{46}$$

The lumped mass of the node is obtained from the summation of all the elements linked at the node. For example, the lumped mass at a node a in element e which has a global index p, (represented as: $(e,a) \in p$)

$$M^p \quad = \quad \sum_{(e,a)\in p} \frac{m^e}{3} \tag{47}$$

Hence, the net Kinetic Energy Integral obtained for the whole mesh would be:

$$K_{n,n+1}(\boldsymbol{x}_n, \boldsymbol{x}_{n+1}) \quad = \quad \sum_p \frac{\Delta t}{2} M^p \;\, \boldsymbol{v}^p_{n+1/2} \cdot \boldsymbol{v}^p_{n+1/2} \tag{48}$$

Thus, the net Lagrangian of the entire mesh becomes:

$$L_{n,n+1}(\boldsymbol{x}_n, \boldsymbol{x}_{n+1}) \quad = \quad \sum_p \frac{\Delta t}{2} M^p \;\, \boldsymbol{v}^p_{n+1/2} \cdot \boldsymbol{v}^p_{n+1/2} \quad - \quad \Delta t\, \Pi(\boldsymbol{x}_n) \tag{49}$$

This leads to the discrete Lagrangian Integral of the Central Difference method, as discussed in [59, 6]. Now the Kinetic Energy Integral is revisited to evaluate the directional derivatives:

$$K_{n,n+1}(\boldsymbol{x}_n, \boldsymbol{x}_{n+1}) \quad = \quad \sum_p \frac{\Delta t}{2} M^p \boldsymbol{v}^p_{n+1/2} \cdot \boldsymbol{v}^p_{n+1/2} \tag{50}$$

where the velocity $\boldsymbol{v}^p_{n+1/2}$ of node p, can be written as:

$$\begin{aligned}\boldsymbol{v}^p_{n+1/2} &= \frac{1}{\Delta t}\left(\boldsymbol{x}^p_{n+1} - \boldsymbol{x}^p_n\right) \\ \frac{\partial K_{n,n+1}}{\partial \boldsymbol{x}^p_n} &= -M^p \boldsymbol{v}^p_{n+1/2} \end{aligned} \tag{51}$$

Similarly :

$$\frac{\partial K_{n-1,n}}{\partial \boldsymbol{x}^p_n} = M^p \boldsymbol{v}^p_{n-1/2} \tag{52}$$

Using equation 17, we obtain the final discrete equation of motion as:

$$\begin{aligned} & \frac{\partial K_{n,n+1}}{\partial \boldsymbol{x}^p_n} + \frac{\partial K_{n-1,n}}{\partial \boldsymbol{x}^p_n} - \Delta t \frac{\partial \Pi^{\mathrm{int}}_n}{\partial \boldsymbol{x}^p_n} - \Delta t \frac{\partial \Pi^{\mathrm{ext}}_n}{\partial \boldsymbol{x}^p_n} \\ = \;& M^p \left(-\boldsymbol{v}^p_{n+1/2} + \boldsymbol{v}^p_{n-1/2}\right) - \Delta t \left(\boldsymbol{T}^p_n - \boldsymbol{F}^p_n\right) \\ = \;& 0 \end{aligned} \tag{53}$$

which can be simplified as:

$$M^p \left(\boldsymbol{v}^p_{n+1/2} - \boldsymbol{v}^p_{n-1/2}\right) = \Delta t \left(\boldsymbol{F}^p_n - \boldsymbol{T}^p_n\right) \tag{54}$$

Thus, we obtain the time integration algorithm of Central Difference Scheme with lumped mass, using linear space-time discretization. The purpose of deriving the commonly known central difference method, was to demonstrate that it belongs to the class of variational integrators [1]. We have proved that the well known lumped mass approximation is a consequence of linear space-time discretization in case of linear triangular elements. In the process of deriving the method, we have also elucidated the use of space-time discretization, which shall be used to develop time-integration updates for time-steps involving mesh changes in the next sections.

3. Mesh Adaptation

In this section, the previously mentioned variational formulation, is extended to mesh adaptation. Mesh adaptations which involve local mesh changes for 2D triangular meshes, are considered. The following operations are formulated separately:

1. Diagonal Swapping.
2. Node Movement.
3. Edge Splitting.
4. Edge Collapsing.

Each of these operations is developed with the assumption that only one of these operations takes place between time level n and $n+1$ on a local patch.

3.1. Diagonal Swapping

A discussion of diagonal swapping is presented, by studying a local patch of two triangular elements abc and acd at time level t_n, as shown in Fig. 3. The patch is time marched to time level t_{n+1} where the common diagonal ac is swapped with the new diagonal bd, thus leading to two different element configurations, abd and bcd at time level t_{n+1}. The space-time volume thus formed, can be subdivided into five tetrahedra: $(a_n b_n c_n b_{n+1})$, $(a_n c_n d_n d_{n+1})$, $(a_{n+1} b_{n+1} d_{n+1} a_n)$, $(b_{n+1} c_{n+1} d_{n+1} c_n)$ and $(a_n c_n b_{n+1} d_{n+1})$ as shown in the figure 3. Note that, the first four tetrahedra, have common nodes, hence the velocity interpolation is simple. The velocity in the fifth(central) tetrahedra is computed by the full expression, (as explained in section 2). Hence, the net Kinetic Energy Integral within the space-time volume can be written as:

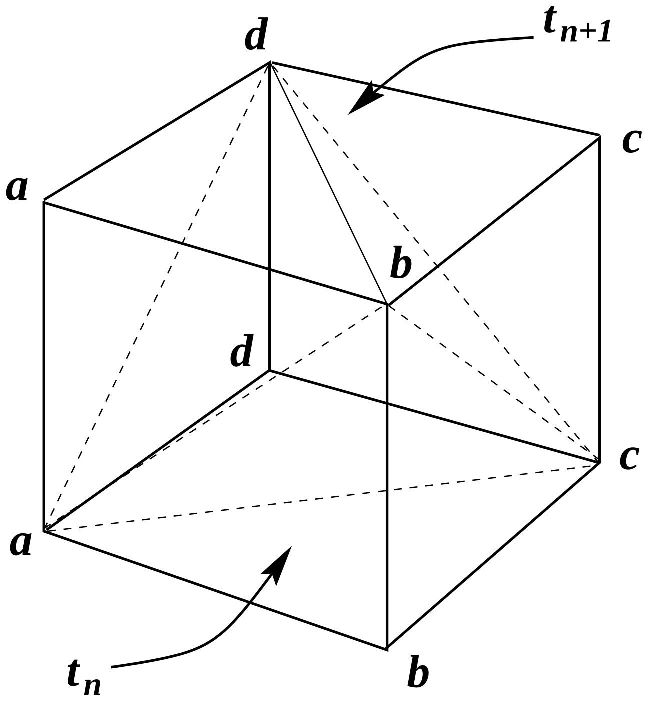

Figure 3. The space-time volume for the diagonal swapping.

$$
\begin{aligned}
K^{abcd}_{n,n+1} \quad &= \quad \frac{\Delta t}{2}\frac{m_{abc}}{3}\boldsymbol{v}^b_{n+1/2}\cdot\boldsymbol{v}^b_{n+1/2} + \frac{\Delta t}{2}\frac{m_{acd}}{3}\boldsymbol{v}^d_{n+1/2}\cdot\boldsymbol{v}^d_{n+1/2} + \frac{\Delta t}{2}\frac{m_{abd}}{3}\boldsymbol{v}^a_{n+1/2}\cdot\boldsymbol{v}^a_{n+1/2} \\
&\quad + \frac{\Delta t}{2}\frac{m_{bcd}}{3}\boldsymbol{v}^c_{n+1/2}\cdot\boldsymbol{v}^c_{n+1/2} + \frac{\Delta t}{2}\frac{m_{abcd}}{3}\boldsymbol{v}^{abcd}_{n+1/2}\cdot\boldsymbol{v}^{abcd}_{n+1/2} \qquad (55) \\
\boldsymbol{v}^{abcd}_{n+1/2} \quad &= \quad \frac{\left(m_{acd}\boldsymbol{x}^b_{n+1} + m_{abc}\boldsymbol{x}^d_{n+1} - m_{bcd}\boldsymbol{x}^a_n + m_{abd}\boldsymbol{x}^c_n\right)}{\Delta t m_{abcd}} \qquad (56)
\end{aligned}
$$

where $m_{abcd} = m_{abc} + m_{acd}$. Using stationarity *wrt.* $\boldsymbol{x}_n$, the contribution to the inertial part of the equilibrium equations at t_n arising from the prism *abcd* is:

$$\begin{aligned} -D_1 K^{abcd}_{n,n+1}[\delta \boldsymbol{x}_n] &= \frac{m_{abc}}{3}\boldsymbol{v}^b_{n+1/2}\cdot\delta\boldsymbol{x}^b_n + \frac{m_{acd}}{3}\boldsymbol{v}^d_{n+1/2}\cdot\delta\boldsymbol{x}^d_n \\ &+ \left(\frac{m_{abd}}{3}\boldsymbol{v}^a_{n+1/2} + \frac{m_{bcd}}{3}\boldsymbol{v}^{abcd}_{n+1/2}\right)\cdot\delta\boldsymbol{x}^a_n \\ &+ \left(\frac{m_{bcd}}{3}\boldsymbol{v}^c_{n+1/2} + \frac{m_{abd}}{3}\boldsymbol{v}^{abcd}_{n+1/2}\right)\cdot\delta\boldsymbol{x}^c_n \end{aligned} \tag{57}$$

Adding this contribution to those arising from non-swapped elements in the mesh leads to an update algorithm at step t_n which for nodes b and d is simply:

$$M^b_n\left(\boldsymbol{v}^b_{n+1/2} - \boldsymbol{v}^b_{n-1/2}\right) = \Delta t\left(\boldsymbol{F}^b_n - \boldsymbol{T}^b_n\right) \tag{58}$$

$$M^d_n\left(\boldsymbol{v}^d_{n+1/2} - \boldsymbol{v}^d_{n-1/2}\right) = \Delta t\left(\boldsymbol{F}^d_n - \boldsymbol{T}^d_n\right) \tag{59}$$

Note that as soon as the position of nodes b and d have been updated, using equations 58 and 59, it is possible to calculate $\boldsymbol{v}^{abcd}_{n+1/2}$ using 56 which in turns allows the update of a and c to take place as:

$$M^a_{n+1}\boldsymbol{v}^a_{n+1/2} - M^a_n\boldsymbol{v}^a_{n-1/2} + \frac{m_{bcd}}{3}\boldsymbol{v}^{abcd}_{n+1/2} = \Delta t\left(\boldsymbol{F}^a_n - \boldsymbol{T}^a_n\right) \tag{60}$$

$$M^c_{n+1}\boldsymbol{v}^c_{n+1/2} - M^c_n\boldsymbol{v}^c_{n-1/2} + \frac{m_{abd}}{3}\boldsymbol{v}^{abcd}_{n+1/2} = \Delta t\left(\boldsymbol{F}^c_n - \boldsymbol{T}^c_n\right) \tag{61}$$

Similarly, using stationarity *wrt.* $\boldsymbol{x}_{n+1}$, the contribution to the equilibrium relations at t_{n+1} is obtained as:

$$\begin{aligned} -D_2 K^{abcd}_{n,n+1}[\delta \boldsymbol{x}_{n+1}] &= \frac{m_{abd}}{3}\boldsymbol{v}^a_{n+1/2}\cdot\delta\boldsymbol{x}^a_{n+1} + \frac{m_{bcd}}{3}\boldsymbol{v}^c_{n+1/2}\cdot\delta\boldsymbol{x}^c_{n+1} \\ &+ \left(\frac{m_{abc}}{3}\boldsymbol{v}^b_{n+1/2} + \frac{m_{acd}}{3}\boldsymbol{v}^{abcd}_{n+1/2}\right)\cdot\delta\boldsymbol{x}^b_{n+1} \\ &+ \left(\frac{m_{acd}}{3}\boldsymbol{v}^d_{n+1/2} + \frac{m_{abc}}{3}\boldsymbol{v}^{abcd}_{n+1/2}\right)\cdot\delta\boldsymbol{x}^d_{n+1} \end{aligned} \tag{62}$$

which lead to an update algorithm at t_{n+1} as:

$$M^a_{n+1}\left(\boldsymbol{v}^a_{n+3/2} - \boldsymbol{v}^a_{n+1/2}\right) = \Delta t\left(\boldsymbol{F}^a_{n+1} - \boldsymbol{T}^a_{n+1}\right) \tag{63}$$

$$M^c_{n+1}\left(\boldsymbol{v}^c_{n+3/2} - \boldsymbol{v}^c_{n+1/2}\right) = \Delta t\left(\boldsymbol{F}^c_{n+1} - \boldsymbol{T}^c_{n+1}\right) \tag{64}$$

$$M^b_{n+1}\boldsymbol{v}^b_{n+3/2} - M^b_n\boldsymbol{v}^b_{n+1/2} - \frac{m_{acd}}{3}\boldsymbol{v}^{n+1/2}_{abcd} = \Delta t\left(\boldsymbol{F}^b_{n+1} - \boldsymbol{T}^b_{n+1}\right) \tag{65}$$

$$M^d_{n+1}\boldsymbol{v}^d_{n+3/2} - M^d_n\boldsymbol{v}^d_{n+1/2} - \frac{m_{abc}}{3}\boldsymbol{v}^{n+1/2}_{abcd} = \Delta t\left(\boldsymbol{F}^d_{n+1} - \boldsymbol{T}^d_{n+1}\right) \tag{66}$$

The momentum within time step t_n and t_{n+1} is ($D_2 L_{n,n+1}$):

$$\boldsymbol{P}_{n,n+1} = \sum_j \boldsymbol{P}^j_{n,n+1} \tag{67}$$

$$\boldsymbol{P}^j_{n,n+1} = \begin{cases} M^j_{n+1}\boldsymbol{v}^j_{n+1/2} & \text{for } j = a \text{ or } c, \\ M^j_n\boldsymbol{v}^j_{n+1/2} + (\frac{m_{acd}}{3})\boldsymbol{v}^{abcd}_{n+1/2} & \text{for } j = b, \\ M^j_n\boldsymbol{v}^j_{n+1/2} + (\frac{m_{abc}}{3})\boldsymbol{v}^{abcd}_{n+1/2} & \text{for } j = d. \end{cases} \tag{68}$$

$$\boldsymbol{H}_{n,n+1} = \sum_j \boldsymbol{x}^j_{n+1}\times\boldsymbol{P}^j_{n,n+1} \tag{69}$$

3.2. Edge Splitting

Now another patch of elements as shown in Fig. 4, is considered to develop the algorithm for edge-splitting. As shown in the figure, a patch of two triangles, abd and bcd, at time level t_n, is time marched to time level t_{n+1}. The common edge bd is split at midpoint e to form four child elements, abe, aed, bce and ecd, at time level t_{n+1}. The space-time volume is now subdivided to five tetrahedra: $(a_n b_n d_n a_{n+1})$, $(b_n c_n d_n c_{n+1})$, $(a_n c_n d_n d_{n+1})$, $(a_n c_n b_n b_{n+1})$, and $(b_n d_n a_{n+1} c_{n+1})$. Note that the first four tetrahedra have common nodes, hence the velocity interpolation is simple. The fifth tetrahedra, is further subdivided into four tetrahedra (as shown in red dotted lines in Fig. 4), each having a common node as e. The point e and the mid point of b and d have the same reference coordinates, $(\boldsymbol{X}_e = \frac{\boldsymbol{X}_b + \boldsymbol{X}_d}{2})$. Thus the Kinetic

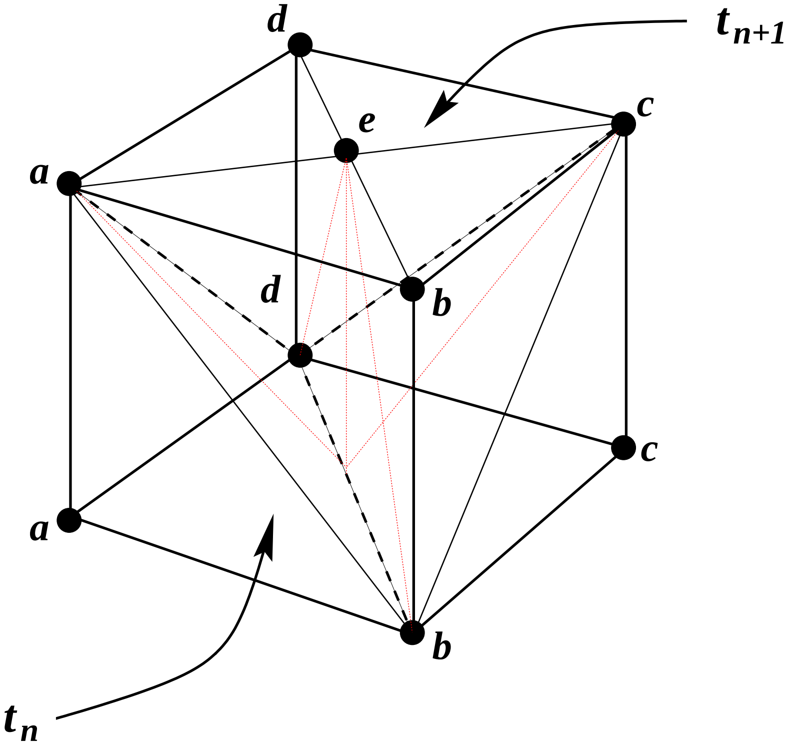

Figure 4. The space-time volume for edge splitting.

Energy Integral can be written as:

$$\begin{aligned} K^{abcd}_{n,n+1} &= \frac{\Delta t}{2}\frac{m_{abd}}{3}\boldsymbol{v}^a_{n+1/2}\cdot\boldsymbol{v}^a_{n+1/2} + \frac{\Delta t}{2}\frac{m_{bcd}}{3}\boldsymbol{v}^c_{n+1/2}\cdot\boldsymbol{v}^c_{n+1/2} \\ &+ \frac{\Delta t}{2}\frac{m_{abe}+m_{bce}}{3}\boldsymbol{v}^b_{n+1/2}\cdot\boldsymbol{v}^b_{n+1/2} + \frac{\Delta t}{2}\frac{m_{dae}+m_{dec}}{3}\boldsymbol{v}^d_{n+1/2}\cdot\boldsymbol{v}^d_{n+1/2} \\ &+ \frac{\Delta t}{2}m_e\boldsymbol{v}^e_{n+1/2}\cdot\boldsymbol{v}^e_{n+1/2} \end{aligned} \tag{70}$$

$$m_e = \frac{(m_{abd}+m_{bcd})}{3} \tag{71}$$

$$\boldsymbol{v}^e_{n+1/2} = \frac{1}{\Delta t}\left[\boldsymbol{x}^e_{n+1} - \frac{\left(\boldsymbol{x}^b_n + \boldsymbol{x}^d_n\right)}{2}\right] \tag{72}$$

Using stationarity *wrt.* $\boldsymbol{x}_n$ one can obtain:

$$\begin{aligned} -D_1K^{abcd}_{n,n+1}[\delta\boldsymbol{x}_n] &= \frac{m_{abd}}{3}\boldsymbol{v}^a_{n+1/2}\cdot\delta\boldsymbol{x}^a_n + \frac{m_{bcd}}{3}\boldsymbol{v}^c_{n+1/2}\cdot\delta\boldsymbol{x}^c_n \\ &+ \frac{m_{abe}+m_{bce}}{3}\boldsymbol{v}^b_{n+1/2}\cdot\delta\boldsymbol{x}^b_n + \frac{m_{dae}+m_{dec}}{3}\boldsymbol{v}^d_{n+1/2}\cdot\delta\boldsymbol{x}^d_n \\ &+ \frac{m_e}{2}\boldsymbol{v}^e_{n+1/2}\cdot\delta\boldsymbol{x}^b_n + \frac{m_e}{2}\boldsymbol{v}^e_{n+1/2}\cdot\delta\boldsymbol{x}^d_n \end{aligned} \tag{73}$$

Thus, the update algorithm at step t_n :

$$M^a_n\left(\boldsymbol{v}^a_{n+1/2} - \boldsymbol{v}^a_{n-1/2}\right) = \Delta t\left(\boldsymbol{F}^a_n - \boldsymbol{T}^a_n\right) \tag{74}$$

$$M^c_n\left(\boldsymbol{v}^c_{n+1/2} - \boldsymbol{v}^c_{n-1/2}\right) = \Delta t\left(\boldsymbol{F}^c_n - \boldsymbol{T}^c_n\right) \tag{75}$$

$$M^b_{n+1}\boldsymbol{v}^b_{n+1/2} - M^b_n\boldsymbol{v}^b_{n-1/2} - \frac{m_e}{2}\boldsymbol{v}^e_{n+1/2} = \Delta t\left(\boldsymbol{F}^a_n - \boldsymbol{T}^a_n\right) \tag{76}$$

$$M^d_{n+1}\boldsymbol{v}^d_{n+1/2} - M^d_n\boldsymbol{v}^d_{n-1/2} - \frac{m_e}{2}\boldsymbol{v}^e_{n+1/2} = \Delta t\left(\boldsymbol{F}^d_n - \boldsymbol{T}^d_n\right) \tag{77}$$

Due to the choice of the mid point on the edge and linear elements, the position of the new node at time level n can be assigned as $\boldsymbol{x}^e_n = \frac{\boldsymbol{x}^b_n+\boldsymbol{x}^d_n}{2}$. Thus the velocity of the new node (also the velocity of the tetrahedron *acbd*) becomes :

$$\boldsymbol{v}^e_{n+1/2} = \frac{1}{2}\left(\boldsymbol{v}^b_{n+1/2} + \boldsymbol{v}^d_{n+1/2}\right) \tag{78}$$

Thus a 2 × 2 system of equation is obtained, to be solved, to obtain the other velocities.

$$\begin{bmatrix} M^b_{n+1} + \frac{1}{4}m_e & \frac{1}{4}m_e \\ \frac{1}{4}m_e & M^d_{n+1} + \frac{1}{4}m_e \end{bmatrix}\begin{bmatrix} \boldsymbol{v}^b_{n+1/2} \\ \boldsymbol{v}^d_{n+1/2} \end{bmatrix} = \begin{bmatrix} M^b_n\boldsymbol{v}^b_{n+1/2} \\ M^d_n\boldsymbol{v}^d_{n+1/2} \end{bmatrix} + \Delta t\begin{bmatrix} \boldsymbol{F}^b_n - \boldsymbol{T}^b_n \\ \boldsymbol{F}^d_n - \boldsymbol{T}^d_n \end{bmatrix} \tag{79}$$

The update at equations at $n+1$ are unchanged. The momentum within time step t_n and t_{n+1} is $(D_2L_{n,n+1})$:

$$\boldsymbol{P}_{n,n+1} = \sum_j \boldsymbol{P}^j_{n,n+1} \tag{80}$$

$$\boldsymbol{P}^j_{n,n+1} = M^j_{n+1}\boldsymbol{v}^j_{n+1/2} \qquad \forall\ j$$

$$\boldsymbol{H}_{n,n+1} = \sum_j \boldsymbol{x}^j_{n+1} \times \boldsymbol{P}^j_{n,n+1} \tag{81}$$

3.3. Node Movement

In order to derive the update equations for node movement, the mapping of the present (spatial) configuration to the reference (material) configuration, is revisited. An arbitrary intermediate configuration(ξ, η) is introduced, as shown in Fig. 5, as is typically done in the case of Arbitrary Lagrangian and Eulerian formulation. The relations between the true and the observed velocity fields can then be written in the following manner:

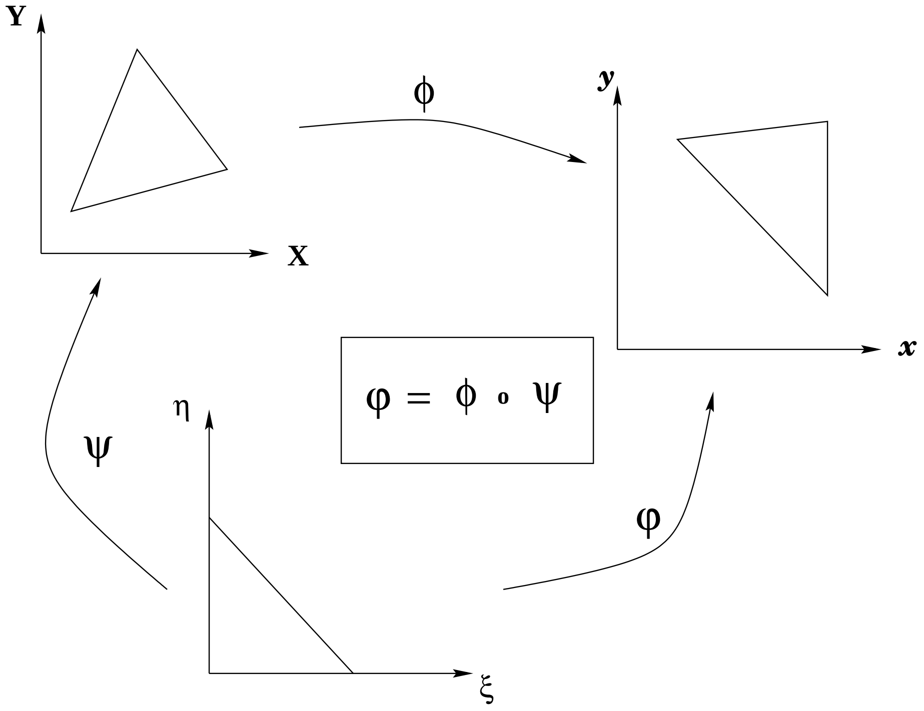

Figure 5. Understanding node movement with an intermediate mapping.

$$\begin{aligned} \text{true velocity}: \ \boldsymbol{v} &= \left.\frac{\partial \boldsymbol{x}}{\partial t}\right|_{\boldsymbol{X}=\text{const.}} &= \frac{\partial}{\partial t}\phi(\boldsymbol{X}, t); \qquad (82)\\ \text{observed velocity}: \ \nu &= \left.\frac{\partial \boldsymbol{x}}{\partial t}\right|_{\xi=\text{const.}} &= \frac{\partial}{\partial t}\varphi(\xi, t); \qquad (83)\\ \text{mesh velocity}: \ \boldsymbol{V} &= \left.\frac{\partial \boldsymbol{X}}{\partial t}\right|_{\xi=\text{const.}} &= \frac{\partial}{\partial t}\psi(\xi, t); \qquad (84) \end{aligned}$$

The true velocity $\boldsymbol{v}$ can be related to the observed velocity ν in terms of the mesh velocity $\boldsymbol{V}$, using the deformation gradient:

$$\begin{aligned} \frac{d\boldsymbol{x}}{dt} &= \frac{\partial \boldsymbol{x}}{\partial t} + \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{X}}\frac{d\boldsymbol{X}}{dt} \\ \nu &= \boldsymbol{v} \ + \ \boldsymbol{F}\boldsymbol{V} \qquad (85) \end{aligned}$$

The Kinetic Energy can then be written as:

$$K_{n,n+1}\left(\boldsymbol{x}_n, \boldsymbol{x}_{n+1}\right) \quad = \quad \frac{\Delta t}{2} \sum_p M^p_{n+1/2} \boldsymbol{v}^p_{n,n+1} \cdot \boldsymbol{v}^p_{n,n+1} \tag{86}$$

where:

$$M^p_{n+1/2} \quad = \quad \frac{1}{2}\left(M^p_n + M^p_{n+1}\right) \tag{87}$$

$$\boldsymbol{v}^p_{n,n+1} \quad = \quad \nu^p_{n+1/2} - \boldsymbol{F}^p_n \boldsymbol{V}^p_{n+1/2} \tag{88}$$

$$\nu^p_{n+1/2} \quad = \quad \frac{1}{\Delta t}\left(\boldsymbol{x}^p_{n+1} - \boldsymbol{x}^p_n\right) \tag{89}$$

$$\boldsymbol{V}^p_{n+1/2} \quad = \quad \frac{1}{\Delta t}\left(\boldsymbol{X}^p_{n+1} - \boldsymbol{X}^p_n\right) \tag{90}$$

$$\boldsymbol{F}^p_n \quad = \quad \left(\frac{\int_{V_0} N^p_n \boldsymbol{F}_n dV}{\int_{V_0} N^p_n dV}\right) \tag{91}$$

The deformation gradient $\boldsymbol{F}_n$ as used in equation 88 is evaluated at time level n in order to make the update explicit. The corresponding equilibrium equations are (for any generic node p, and its neighboring nodes s):

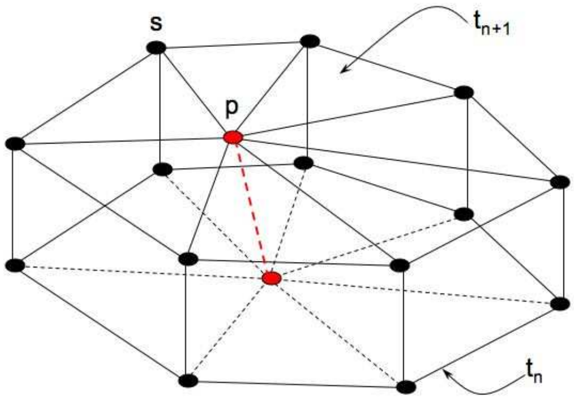

Figure 6. Space-time volume for node movement.

$$M^p_{n+1/2} \boldsymbol{v}^p_{n,n+1} - M^p_{n-1/2} \boldsymbol{v}^p_{n-1,n} \quad = \quad -\Delta t \boldsymbol{Q}^p_{n,n+1} + \Delta t \left(\boldsymbol{F}^p_n - \boldsymbol{T}^p_n\right) \tag{92}$$

where:

$$\boldsymbol{v}^p_{n,n+1} = \frac{1}{\Delta t}\left[\left(\boldsymbol{x}^p_{n+1} - \boldsymbol{x}^p_n\right) - \boldsymbol{F}^p_n\left(\boldsymbol{X}^p_{n+1} - \boldsymbol{X}^p_n\right)\right] \tag{93}$$

$$\boldsymbol{Q}^p_{n,n+1} = \sum_{a=p,s} \frac{m^a_{n+1/2}}{m^a_n}\left(\boldsymbol{v}^a_{n,n+1} \otimes \boldsymbol{V}^a_{n,n+1}\right)\int_{V_0} \rho_0 N^a_n \nabla_0 N^p_n dV \tag{94}$$

In the case where neighboring nodes, s, remain fixed i.e. ($\boldsymbol{V}^s_{n,n+1} = 0$) and the total patch volume V_0 remains fixed (implying $\frac{m^p_{n+1/2}}{m^p_n} = 1$), the expression for $\boldsymbol{Q}^p_{n,n+1}$ becomes:

$$\boldsymbol{Q}^p_{n,n+1} = \left(\boldsymbol{v}^p_{n,n+1} \otimes \boldsymbol{V}^p_{n,n+1}\right)\int_{V_0} \rho_0 N^p_n \nabla_0 N^p_n dV = 0$$

Since the integral $\int_{V_0} \rho_0 N^p_n \nabla_0 N^p_n dV$ becomes zero for an internal node, for uniform density. Thus the update step for the node to be moved (p) becomes:

$$M^p_{n+1/2}\boldsymbol{v}^p_{n,n+1} - M^p_{n-1/2}\boldsymbol{v}^p_{n-1,n} = \Delta t\left(\boldsymbol{F}^p_n - \boldsymbol{T}^p_n\right) \tag{95}$$

And for the neighboring nodes (s), the update step becomes:

$$M^s_{n+1/2}\boldsymbol{v}^s_{n,n+1} - M^s_{n-1/2}\boldsymbol{v}^s_{n-1,n} = -\Delta t\boldsymbol{Q}^s_{n,n+1} + \Delta t\left(\boldsymbol{F}^s_n - \boldsymbol{T}^s_n\right) \tag{96}$$

$$\boldsymbol{Q}^s_{n,n+1} = \left(\boldsymbol{v}^p_{n,n+1} \otimes \boldsymbol{V}^p_{n,n+1}\right)\int_{V_0} \rho_0 N^p_n \nabla_0 N^s_n dV \tag{97}$$

The momentum within time step t_n and t_{n+1} is $(D_2 L_{n,n+1})$:

$$\boldsymbol{P}_{n,n+1} = \sum_j \boldsymbol{P}^j_{n,n+1} \tag{98}$$

$$\boldsymbol{P}^j_{n,n+1} = M^{n+1/2}_j \boldsymbol{v}^j_{n+1/2} \quad \forall\, j$$

$$\boldsymbol{H}_{n,n+1} = \sum_j \boldsymbol{x}^j_{n+1} \times \boldsymbol{P}^j_{n,n+1} \tag{99}$$

3.4. Edge Collapsing

Edge collapsing operation is approached by visualizing a generic patch of elements, as shown in Fig. 7. In the triangular element *arb*, the edge *ab* is wished to be collapsed, leading to removal of the triangles *arb* and *abs*. The points *a* and *b*, belonging to time level *n*, is substituted by the new point *c* at time level $n + 1$ as shown.

The space-time volume as shown in Fig. 8 is the volume over which the Lagrangian is to be computed. To do so, the space-time volume is sub-divided into tetrahedra. There are mainly three types of tetrahedra as shown in Fig. 9. The first type (I) encloses the volume *arbsc.* Then based on the surrounding nodes there are two types of tetrahedra, as shown in Fig. 9. The tetrahedra having *a* or *b* as one of the vertices and the surrounding nodes from time level n and $n + 1$, are labeled type (II). The tetrahedra having *c* as one of their vertices and the surrounding nodes from the time level $n + 1$ and *a* or *b* as the fourth node, are labeled type (III). The location of the new node *c* is chosen to be a linear interpolation of the locations of nodes *a*,*b* and *r*.

$$\boldsymbol{X}^c_{n+1} = \xi\boldsymbol{X}^a_n + \eta\boldsymbol{X}^b_n + (1 - \xi - \eta)\boldsymbol{X}^r_n \tag{100}$$

$$\boldsymbol{x}^c_n = \xi\boldsymbol{x}^a_n + \eta\boldsymbol{x}^b_n + (1 - \xi - \eta)\boldsymbol{x}^r_n \tag{101}$$

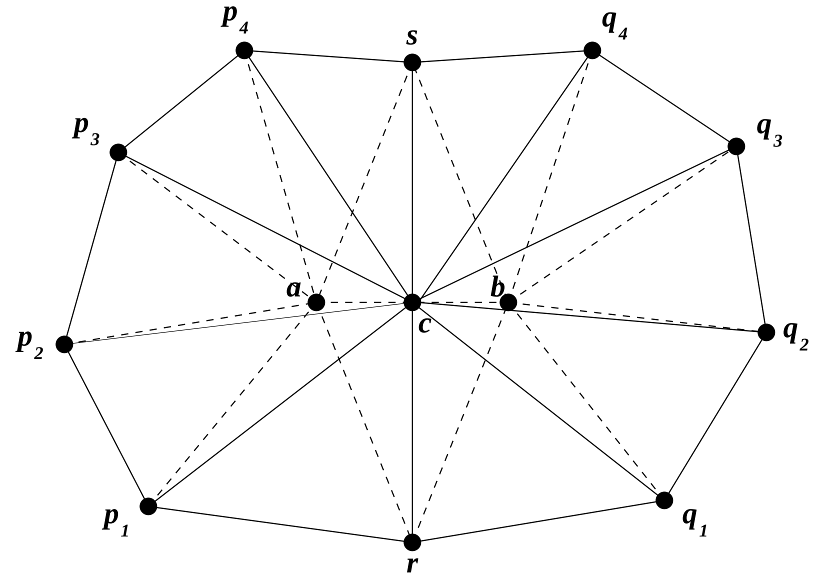

Figure 7. Collapsing the edge ab to the point c.

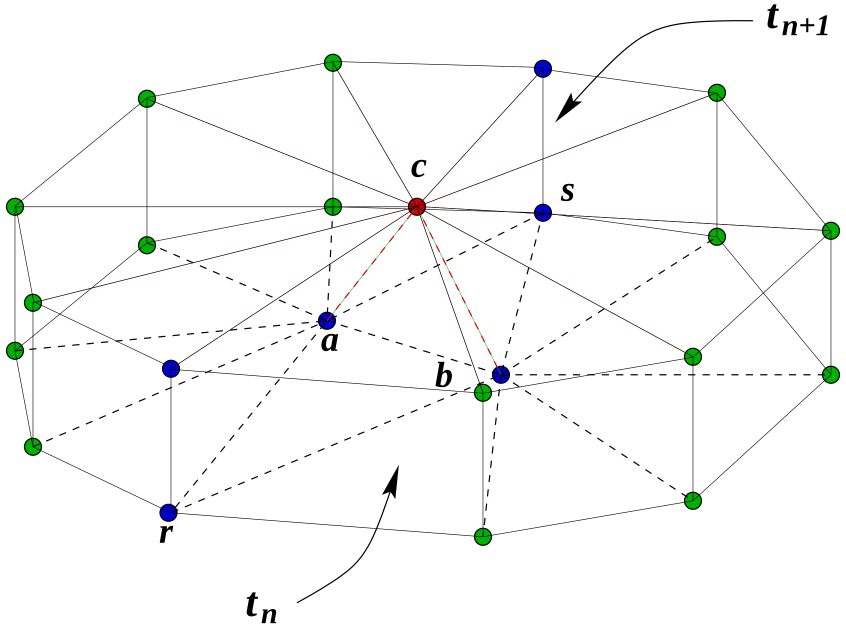

Figure 8. The space-time volume for Edge collapsing operation.

The Kinetic Energy Integral and the velocity interpolation within the tetrahedra of type (I)

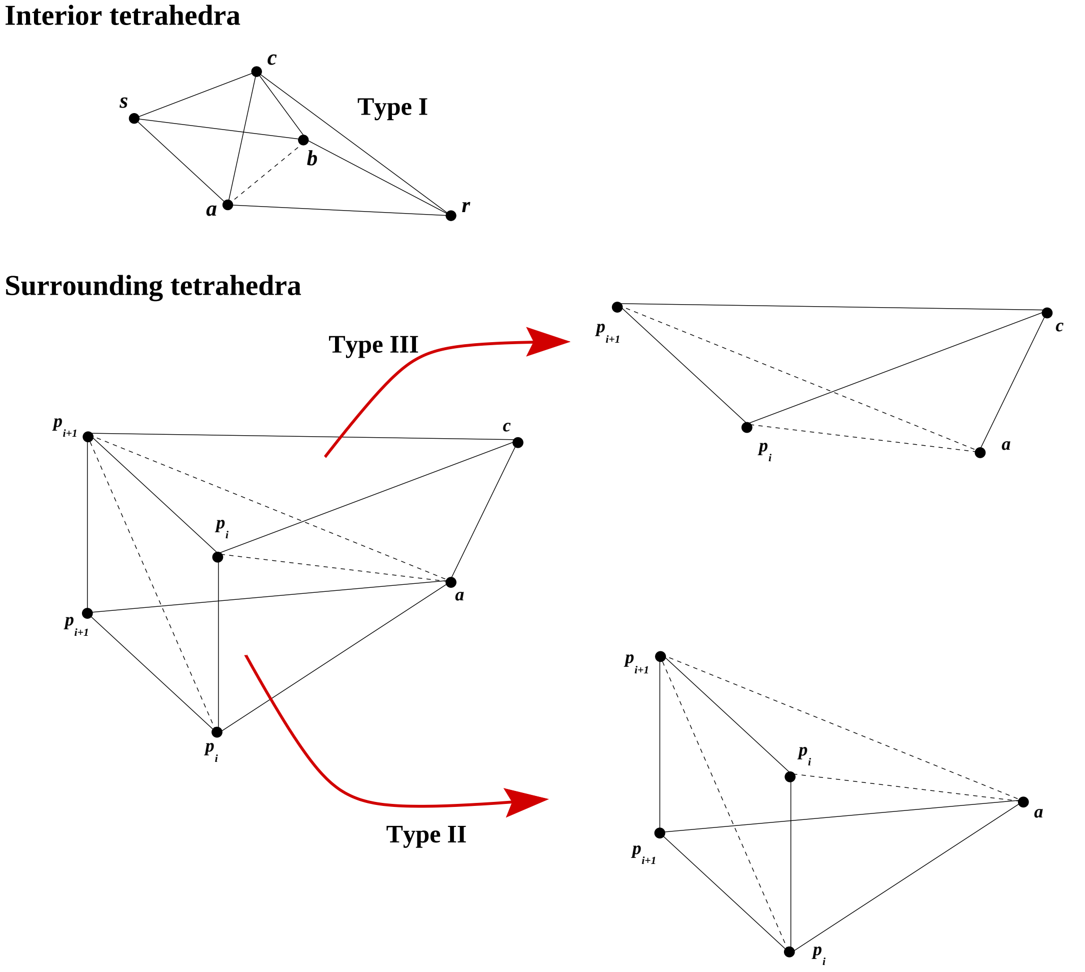

Figure 9. The subdivision of the space-time volume into different types of tetrahedra.

can be written as follows:

$$\begin{aligned} K^{I}_{n,n+1} &= \frac{\Delta t}{2}\frac{m^{n}_{ab}}{3}\boldsymbol{v}^{c}_{n+1/2}\cdot\boldsymbol{v}^{c}_{n+1/2} && (102)\\ m^{n}_{ab} &= m^{n}_{arb} + m^{n}_{abs} \\ v^{c}_{n+1/2} &= \frac{\boldsymbol{x}^{c}_{n+1} - (\xi\boldsymbol{x}^{a}_{n} + \eta\boldsymbol{x}^{b}_{n} + (1-\xi-\eta)\boldsymbol{x}^{r}_{n})}{\Delta t} && (103) \end{aligned}$$

Similarly the Kinetic Energy Integral and the velocity interpolation within the tetrahedra of type (II) can be written as:

$$\begin{aligned} K^{II}_{n,n+1} &= \frac{\Delta t}{2} m^{n}_{g_i}\boldsymbol{v}^{g_i}_{n+1/2}\cdot\boldsymbol{v}^{g_i}_{n+1/2} && (104)\\ v^{g_i}_{n+1/2} &= \frac{\boldsymbol{x}^{g_i}_{n+1} - \boldsymbol{x}^{g_i}_{n}}{\Delta t} && (105) \end{aligned}$$

where the index g_i is the overall index of all the neighboring nodes, ordered as $(g_i = r, q_i, s, p_i)$.

The velocity in the tetrahedra of type (III) is not straight forward, since there is no common node in each tetrahedron. Hence the full expression of the velocity (described previously) is used.

$$\begin{aligned} v^{g_i g_{i+1} ac}_{n+1/2} &= \frac{A_{g_{i+1}ac}\boldsymbol{x}^{g_i}_{n+1} + A_{g_i ac}\boldsymbol{x}^{g_{i+1}}_{n+1} + A_{g_i c g_{i+1}}\boldsymbol{x}^{c}_{n+1} - A_{g_i a g_{i+1}}\boldsymbol{x}^{a}_{n}}{6\mathcal{V}_{g_i g_{i+1} ac}} && (106) \\ K^{g_i(III)}_{n,n+1} &= m^n_{g_i g_{i+1} ac}\boldsymbol{v}^{g_i g_{i+1} ac}_{n+1/2} \cdot \boldsymbol{v}^{g_i g_{i+1} ac}_{n+1/2} && (107) \end{aligned}$$

This makes the algorithm very complex. In order to simplify the algorithm, an approximation is made. The Kinetic Energy Integral from each of the tetrahedra, of type (III), are added together, and the sum is expressed by the following approximation :

$$\begin{aligned} \sum_{g_i} K^{g_i(III)}_{n,n+1} &= \frac{\Delta t}{2} m^{n+1}_c \boldsymbol{v}^*_{n+1/2} \cdot \boldsymbol{v}^*_{n+1/2} && (108) \\ \boldsymbol{v}^*_{n+1/2} &= \frac{1}{\Delta t m^{n+1}_c}\left[m^*_c \boldsymbol{x}^c_{n+1} + \sum_i \Delta m_{g_i}\boldsymbol{x}^{g_i}_{n+1} - m^*_a \boldsymbol{x}^a_n - m^*_b \boldsymbol{x}^b_n \right] && (109) \\ m^*_c &= m^{n+1}_c - \frac{m^n_{ab}}{3} && (110) \\ \Delta m_{g_i} &= m^{n+1}_{g_i} - m^n_{g_i} && (111) \\ m^*_a &= m^n_a - \frac{m^n_{ab}}{3} + M_{arcs} && (112) \\ m^*_b &= m^n_b - \frac{m^n_{ab}}{3} + M_{bscr} && (113) \end{aligned}$$

Where M_{arcs} and M_{bscr} are the masses enclosed within *arcs* and *bscr* respectively. Note that Δm_{g_i}, m^*_a and m^*_b can be expressed as linear functions of ξ and η. The velocity $\boldsymbol{v}^*_{n+1/2}$ is a weighted average of the velocities of all the tetrahedra of type III as calculated in equation 106. In addition, since the neighboring nodes are not moved, nor are any neighboring edge allowed to be collapsed, the mass m^{n+1}_c is known *apriori*. Thus the net Kinetic Energy integral becomes:

$$\begin{aligned} K_{n,n+1} &= K^I_{n,n+1} + K^{II}_{n,n+1} + K^{III}_{n,n+1} \\ &= \frac{\Delta t}{2}\frac{m^n_{ab}}{3}\boldsymbol{v}^c_{n+1/2} \cdot \boldsymbol{v}^c_{n+1/2} \quad + \quad \frac{\Delta t}{2} m^{n+1}_c \boldsymbol{v}^*_{n+1/2} \cdot \boldsymbol{v}^*_{n+1/2} \\ &\quad + \quad \sum_i \frac{\Delta t}{2} m^n_{g_i}\boldsymbol{v}^{g_i}_{n+1/2} \cdot \boldsymbol{v}^{g_i}_{n+1/2} && (114) \end{aligned}$$

Using stationarity *wrt.* $\boldsymbol{x}_n$, the equilibrium equations obtained at time level n are as follows:

$$\begin{aligned} (\forall\ g_i \neq r) \qquad m^n_{g_i}\ \boldsymbol{v}^{g_i}_{n+1/2} - m^n_{g_i}\ \boldsymbol{v}^{g_i}_{n-1/2} &= \Delta t\ (\boldsymbol{F}^{g_i}_n - \boldsymbol{T}^{g_i}_n) && (115) \\ (1-\xi-\eta)\ \frac{m^n_{ab}}{3}\ \boldsymbol{v}^c_{n+1/2} \ + \ m^n_r\ \boldsymbol{v}^r_{n+1/2} \ - \ m^n_r\ \boldsymbol{v}^r_{n-1/2} &= \Delta t\ (\boldsymbol{F}^r_n - \boldsymbol{T}^r_n) && (116) \\ \xi\ \frac{m^n_{ab}}{3}\ \boldsymbol{v}^c_{n+1/2} \ + \ m^*_a\ \boldsymbol{v}^*_{n+1/2} \ - \ m^n_a\ \boldsymbol{v}^a_{n-1/2} &= \Delta t\ (\boldsymbol{F}^a_n - \boldsymbol{T}^a_n) && (117) \\ \eta\ \frac{m^n_{ab}}{3}\ \boldsymbol{v}^c_{n+1/2} \ + \ m^*_b\ \boldsymbol{v}^*_{n+1/2} \ - \ m^n_b\ \boldsymbol{v}^b_{n-1/2} &= \Delta t\ \left(\boldsymbol{F}^b_n - \boldsymbol{T}^b_n\right) && (118) \end{aligned}$$

Here, a new variable $\boldsymbol{R}_n^j$ is introduced, where

$$(\forall\ j = g_i, a, b) \qquad \boldsymbol{R}_n^j \;=\; m_j^n\ \boldsymbol{v}_{n-1/2}^j + \Delta t\ \left(\boldsymbol{F}_n^j - \boldsymbol{T}_n^j\right) \tag{119}$$

Note that $\boldsymbol{R}_n^j$ is known *apriori*. Hence the set of equations, can be rewritten as:

$$\begin{aligned}
(\forall\ g_i \neq r) \qquad m_{g_i}^n\ \boldsymbol{v}_{n+1/2}^{g_i} &= \boldsymbol{R}_n^{g_i} && (120)\\
(1-\xi-\eta)\ \frac{m_{ab}^n}{3}\ \boldsymbol{v}_{n+1/2}^c\ +\ m_r^n\ \boldsymbol{v}_{n+1/2}^r &= \boldsymbol{R}_n^r && (121)\\
\xi\ \frac{m_{ab}^n}{3}\ \boldsymbol{v}_{n+1/2}^c\ +\ m_a^*\ \boldsymbol{v}_{n+1/2}^* &= \boldsymbol{R}_n^a && (122)\\
\eta\ \frac{m_{ab}^n}{3}\ \boldsymbol{v}_{n+1/2}^c\ +\ m_b^*\ \boldsymbol{v}_{n+1/2}^* &= \boldsymbol{R}_n^b && (123)
\end{aligned}$$

Note here that Eqn. 120 is fully explicit, hence, $\boldsymbol{x}_{n+1}^{g_i}$ for all g_i except r are known. Now revisiting Eqn. 109 one can rewrite the expression for $\boldsymbol{v}_{n+1/2}^*$ using Eqn. 121 as:

$$\begin{aligned}
\boldsymbol{v}_{n+1/2}^* &= S_m(\xi,\eta)\boldsymbol{v}_{n+1/2}^c + \boldsymbol{W}(\xi,\eta) && (124)\\
\boldsymbol{W}(\xi,\eta) &= \frac{1}{\Delta t m_c^{n+1}}\left[\sum_i \Delta m_{g_i}\boldsymbol{z}_{n+1}^{g_i} - m_a^*\boldsymbol{x}_n^a - m_b^*\boldsymbol{x}_n^b + m_c^*\boldsymbol{x}_n^c + \Delta m_r\boldsymbol{x}_n^r\right]\\
&= \boldsymbol{W}_0 + \xi\ \boldsymbol{W}_\xi + \eta\ \boldsymbol{W}_\eta && (125)
\end{aligned}$$

where,

$$\begin{aligned}
\boldsymbol{z}_{n+1}^{g_i} &= \boldsymbol{x}_{n+1}^{g_i} && (\forall\ g_i \neq r)\\
&= \frac{\Delta t\ \boldsymbol{R}_n^r}{m_n^r} && (\ g_i = r)\\
S_m(\xi,\eta) &= \frac{m_{ab}^n}{3m_{n+1}^c m_n^r}\left[\frac{3m_c^* m_n^r}{m_{ab}^n} - (\Delta m_r(1-\xi-\eta))\right]\\
&= S_0 + S_\xi^1\xi + S_\eta^1\eta + S_\xi^2\xi^2 + S_\eta^2\eta^2 + S_{\xi\eta}^2\xi\eta && (126)
\end{aligned}$$

Note that the vector coefficients ($\boldsymbol{W}_0$, $\boldsymbol{W}_\xi$, $\boldsymbol{W}_\eta$) and the scalar coefficients ($S_0, S_\xi^1, S_\eta^1, S_\xi^2, S_\eta^2, S_{\xi\eta}^2$) are known *apriori*. Using Eqn. 124 in Eqns. 122 & 123 the two equations are rewritten as:

$$\begin{aligned}
K_a(\xi,\eta)\ \boldsymbol{v}_{n+1/2}^c\ +\ m_a^*(\xi,\eta)\ \boldsymbol{W}(\xi,\eta) &= \boldsymbol{R}_n^a && (127)\\
K_b(\xi,\eta)\ \boldsymbol{v}_{n+1/2}^c\ +\ m_b^*(\xi,\eta)\ \boldsymbol{W}(\xi,\eta) &= \boldsymbol{R}_n^b && (128)
\end{aligned}$$

where,

$$\begin{aligned}
K_a(\xi,\eta) &= \xi\ \frac{m_{ab}^n}{3} + m_a^*(\xi,\eta)\ S_m(\xi,\eta) && (129)\\
K_b(\xi,\eta) &= \eta\ \frac{m_{ab}^n}{3} + m_b^*(\xi,\eta)\ S_m(\xi,\eta) && (130)
\end{aligned}$$

Now, eliminating $\boldsymbol{v}_{n+1/2}^c$ from both the above equations, the following equations are obtained:

$$\begin{aligned}
\boldsymbol{v}_{n+1/2}^c &= \frac{\boldsymbol{R}_n^a - m_a^*(\xi,\eta)\ \boldsymbol{W}(\xi,\eta)}{K_a(\xi,\eta)} && (131)\\
\boldsymbol{f}(\xi,\eta) &\equiv \frac{K_b}{K_a}\left(\boldsymbol{R}_n^a -\ m_a^*\ \boldsymbol{W}\right) + \left(\ m_b^*\ \boldsymbol{W} - \boldsymbol{R}_n^b\right) \;=\; 0 && (132)
\end{aligned}$$

Thus, a simple vector equation (132) is obtained, which is used to determine the scalars ξ and η by which, the position of the new node c is determined. This is a coupled quadratic equation which is solved by iteration. A simple Newton iteration leads to quadratic convergence. This leads to the position of the new node ($\boldsymbol{X}^c_{n+1}$) to be a solution of the local equilibrium. Edge ab is collapsed only if the node c lies within the area included by all the surrounding nodes g_i.

Once the position of the node c is obtained, the velocity updates are obtained through simple explicit equations mentioned above (131,124 and 121). The position updates are obtained by the Eqns. 103 & 105. The momentum conserved in this time-step is of the form:

$$\boldsymbol{P}_{n,n+1} = \sum_j \boldsymbol{P}^j_{n,n+1} \tag{133}$$

$$\boldsymbol{P}^j_{n,n+1} = \begin{cases} m^j_n \boldsymbol{v}^j_{n+1/2} + \Delta m_j \boldsymbol{v}^*_{n+1/2}, & \text{for } j = g_i \\ \frac{1}{3} m^n_{ab} \boldsymbol{v}^j_{n+1/2} + (m^j_{n+1} - \frac{m^n_{ab}}{3}) \boldsymbol{v}^*_{n+1/2}, & \text{for } j = c \end{cases} \tag{134}$$

$$\boldsymbol{H}_{n,n+1} = \sum_j \boldsymbol{x}^j_{n+1} \times \boldsymbol{P}^j_{n,n+1} \tag{135}$$

Similar to the previous time-step, using stationarity *wrt.* $\boldsymbol{x}_{n+1}$, the equilibrium equations for the next time step t_{n+1} are obtained. The final update equations are:

$$m^{g_i}_{n+1} \boldsymbol{v}^{g_i}_{n+1/2} - m^{g_i}_n \boldsymbol{v}^{g_i}_{n+1/2} - \Delta m_{g_i} \boldsymbol{v}^*_{n+1/2} = \Delta t \, (\boldsymbol{F}^{g_i}_{n+1} - \boldsymbol{T}^{g_i}_{n+1}) \tag{136}$$

$$m^c_{n+1} \boldsymbol{v}^c_{n+1/2} - \frac{1}{3} m^n_{ab} \boldsymbol{v}^c_{n+1/2} - (m^c_{n+1} - \frac{m^n_{ab}}{3}) \boldsymbol{v}^*_{n+1/2} = \Delta t \, (\boldsymbol{F}^c_{n+1} - \boldsymbol{T}^c_{n+1}) \tag{137}$$

4. Error Estimate and Adaptation Criteria

Using the mesh adaptation procedures, explained so far, an effective mesh-adaptive solver can be implemented which is momentum conserving. In order to develop a mesh adaptive solver, a suitable mesh adaptation criteria, based on error estimates was used. A gradient-type error estimate described by Zienkiewicz and Zhu,[46, 47] (commonly known as Z^2 error-estimate) was used. The stresses in each element, and its neighboring elements, were used to obtain a "recovered stress" at the element. The difference of these two stresses leads to the error estimate at the element. The details of this error estimate can be found in [59]. Elements with high values of this error, were chosen for adaptation.

Mesh adaptation was performed as a sequence of all the mesh operations (diagonal swapping, edge splitting, edge collapsing or node-movement) in tandem. Each mesh operation involved two timesteps ($t_n \rightarrow t_{n+1}$) and ($t_{n+1} \rightarrow t_{n+2}$). Only one type of mesh operation was attempted within each pair of timesteps, over the whole mesh. Mesh operations were attempted after constant intervals (number of time-steps).

At the beginning of each mesh operation, a simple non-adaptive update was conducted. Then based on the error-estimate, elements were chosen. Using the element connectivity information, a local patch, associated with the chosen element was identified. Any other element touching this patch (sharing a common node) was restricted from that mesh-operation in that time-step. Within the element, edge-length-ratio of each edge (η_i) was obtained by the relation $\left(\eta_i = \frac{2l_i}{\sum_j l_j}\right)$, l_i being the length of the i^{th} edge. The edge-length-ratios varied from 0 to 1.

Values close to zero or one, indicated distorted elements. Edges, with such extreme values of edge-length-ratios, were collapsed, split or swapped. Edges with edge-length-ratios close to 0, were collapsed. Edges with edge-length-ratios close to 1, were swapped or split. The minimum edge-length-ratio for collapsing or the maximum edge-length-ratio for swapping or splitting were threshold values chosen for each problem.

In case of node movement, a local patch of nodes were considered, and the average (centroid) location of the nodes and the deviation of the node from the average location was calculated. For higher deviation values, the node was moved towards the centroid.

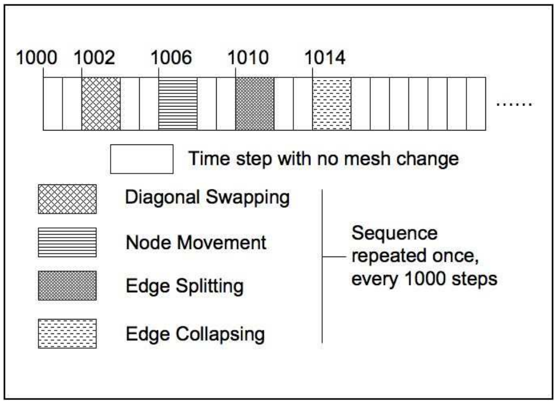

Figure 10. Mesh adaptation operations conducted in tandem, each operation requiring a pair of time-steps.

Diagonal swapping, node movement, edge-splitting and edge-collapsing were attempted in this sequence at every subsequent (or alternate) timestep pair, as shown in figure 10. Based on resolution requirements, the lowest (finest) hierarchical level of the grid was prescribed, in order to prevent over-refinement. The zero'th (coarsest) hierarchical level elements were not removed, in order to prevent over-coarsening. Typically the threshold values for edge-length ratios, maximum hierarchical levels and the frequency of adaptation were specified as input parameters for each operation.

The adaptation criteria used for the present mesh operations were chosen for their simplicity of implementation. Further development of the mesh adaptation criteria, is required for generic cases. In the next section, performance of the final mesh-adaptive solver is demonstrated using examples from rapid dynamics of hyperelastic bodies. The purpose behind these examples is to demonstrate that the updates developed in the previous sections are momentum conserving. Such mesh adaptive time updates can be implemented with different mesh-adaptation criteria, with very less additional computational expense.

Time integration was performed such that the entire mesh was kept at the same time level (synchronous integration). The critical characteristic mesh size, h, was chosen as the smallest radius of the inscribed circle in each element. The time step was calculated based on the characteristic mesh size, maximum wave speed ($v_{\max} = v + a$, with a being the fastest wave speed in the material and v is the maximum velocity in the mesh) and the CFL (Courant-

Friedrichs-Lewy) number, ν, as:

$$\Delta t = \nu \frac{h}{v_{\max}} \tag{138}$$

CFL number's typically ranged for 0.1 to 0.4 for the following examples. Lower CFL numbers are commonly encountered in more complex rapid dynamics problems.

5. Examples

5.1. *Spinning Plate*

A unit thickness square plate, spinning without any constraint, was considered as a test case to illustrate the conservation properties of the proposed mesh adaptation procedures. The plate was made out of nearly incompressible rubber material with material properties, *viz.*, Young's Modulus $E = 1.7 \times 10^7$ Pa, Poisson's ratio $\nu = 0.45$ and density $\rho = 1.1 \times 10^3\,\mathrm{kg/m^3}$. The plate rotated at 1000 RPM. The plate was meshed with 200 equal linear triangular elements as shown in Figure 11 (left) which also shows the pressure distribution at a given instant (right). The simulation was conducted using a mesh-adaptive solver, using all the above mentioned, adaptation procedures *viz.*, diagonal swapping, node movement, edge-splitting and edge collapsing as explained in the previous section. Figure 11, demonstrates how the mesh was refined where the mesh skewness and stresses were relatively larger. The linear momentum $\boldsymbol{P}_{n,n+1}$ and angular momentum $\boldsymbol{H}_{n,n+1}$ are expected to remain constant during the time integration. The center of mass was initially at $\boldsymbol{X} = 0.5$, $\boldsymbol{Y} = 0.5$, and as a consequence of conservation of linear momentum $\boldsymbol{P}_{n,n+1}$, remains at the same location.

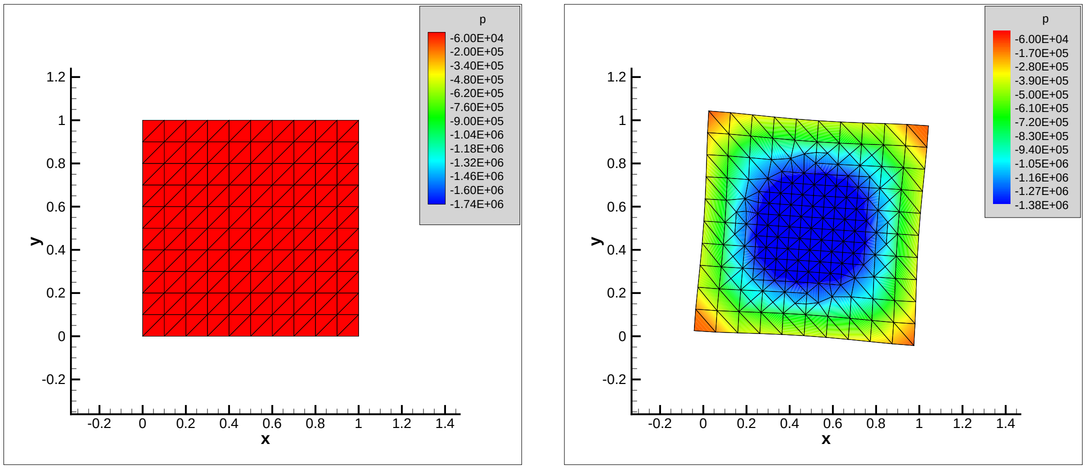

Figure 11. Example 5.1 : A Spinning plate simulation with adaptation.

The momentum remains conserved exactly, throughout the simulation as shown in figure 12. A zoomed in plot depicts the variations of momentum at a smaller scale, which are of the order of the machine precision and remains bounded over the entire simulation. The momentum

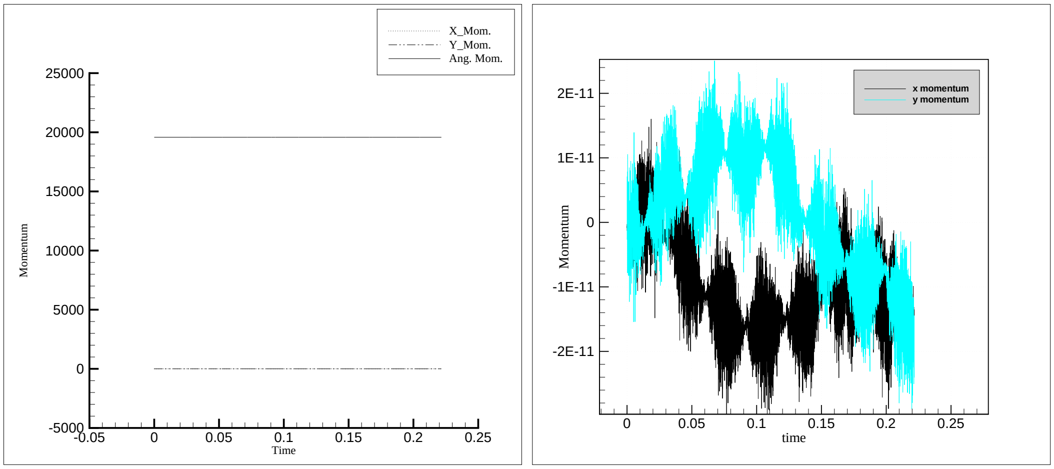

Figure 12. Example 5.1: Linear and angular momentum history. The figure at the right shows the same plot of linear momentum at a smaller scale.

calculated in each step was based on the $\boldsymbol{P}_{n,n+1}$ and $\boldsymbol{H}_{n,n+1}$ expressions described in each of the adaptation procedures previously. The center of mass remains at (0.5,0.5) as shown in figure 13. Viscous stabilization as described in [59], was used to dissipate error accumulation over several time integration steps. Energy history, as shown in figure 14, shows very less energy dissipation in this case. Figure 14, also shows the adaptation history, with the cumulative number of operations conducted and the total number of nodes and elements at each instance of time during the course of adaptation. Figure 15 shows the behavior of the error (as compared to the error in the finer mesh) with mesh size and the relative computational time involved. The mesh size for the adapted solution was taken as the mean of the coarsest and finest mesh

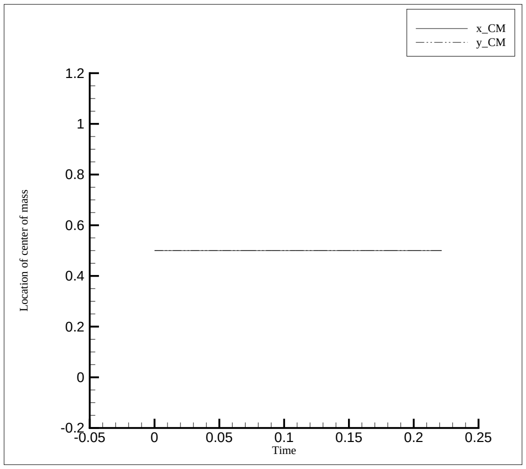

Figure 13. Example 5.1 : Location of center of mass.

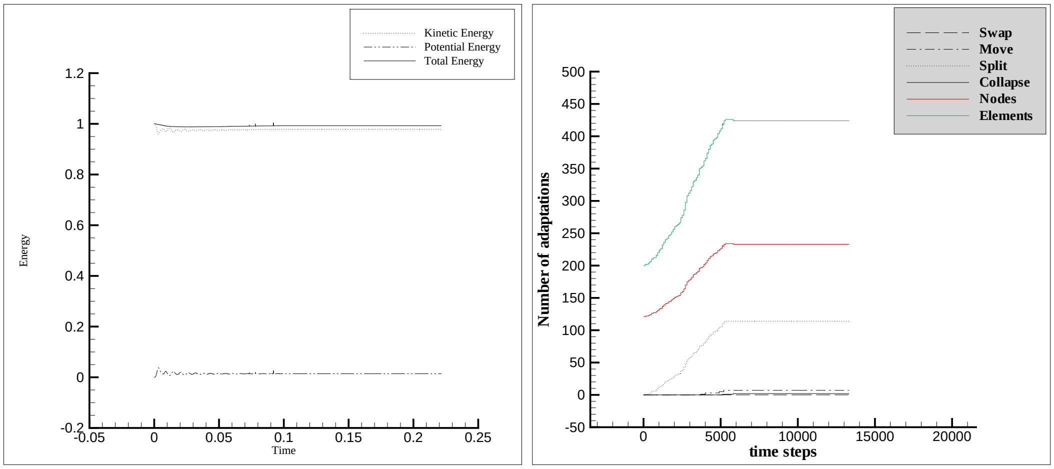

Figure 14. Example 5.1 : Energy history (left) and Adaptation History (right).

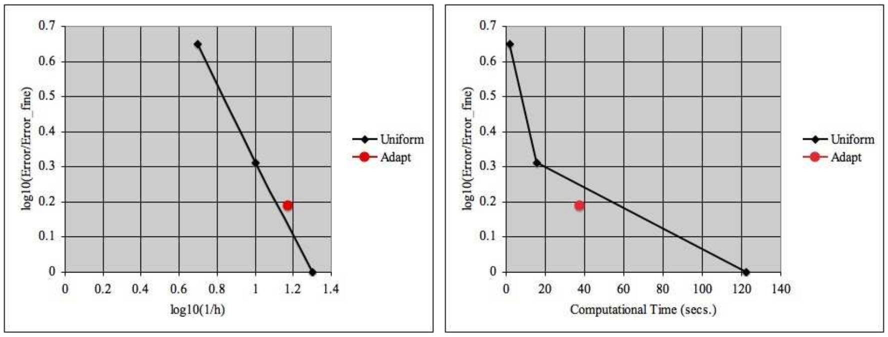

Figure 15. Example 5.1 : The figure in the left shows the error behavior with mesh size. The figure in right shows the computational times for each of the computations.

size within the adapted solution. It is shown through figure 15 (right), that error-reduction is accomplished using mesh-adaptation with low additional computational costs.

5.2. An oscillating ring

A unit thickness circular ring, made up of nearly incompressible hyperelastic material (rubber) ($E = 1.7 \times 10^7$ Pa, $\nu = 0.48$ and $\rho = 10^3$ kg/m^3) is initially stretched to 1.5 its diameter and thereafter let to oscillate freely. This was chosen as another test case to study the momentum conservation property of mesh adaptation. The initial configuration of the ring is shown in figure 16 (left). The ring was stretched as shown in figure 16 (right), at time $t = 0$ and thereafter let to oscillate freely. The simulation was performed up to time $t = 0.2s$, involving 24410 time steps, with mesh adaptations at every 1000 steps.

Figure 17 - 18 show the spatial (deformed) configurations at intermediate time steps. Linear and Angular momentum remains conserved throughout the simulation. The center of mass

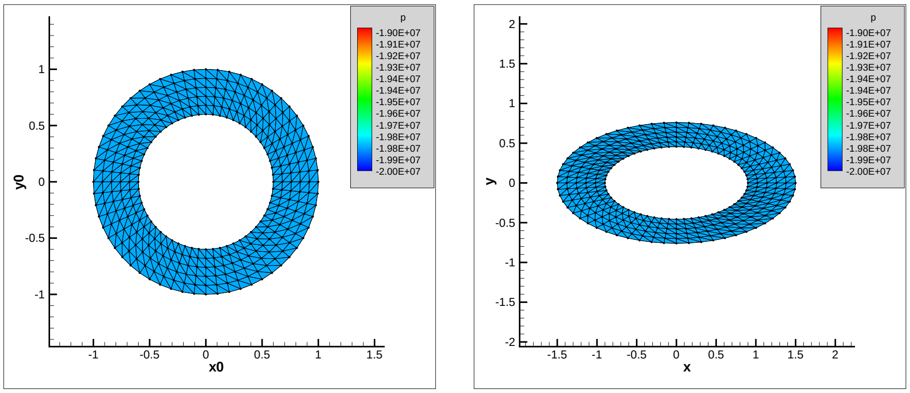

Figure 16. Example 5.2 : Ring at time $t = 0\,s$, with material(left) and spatial(right) configurations.

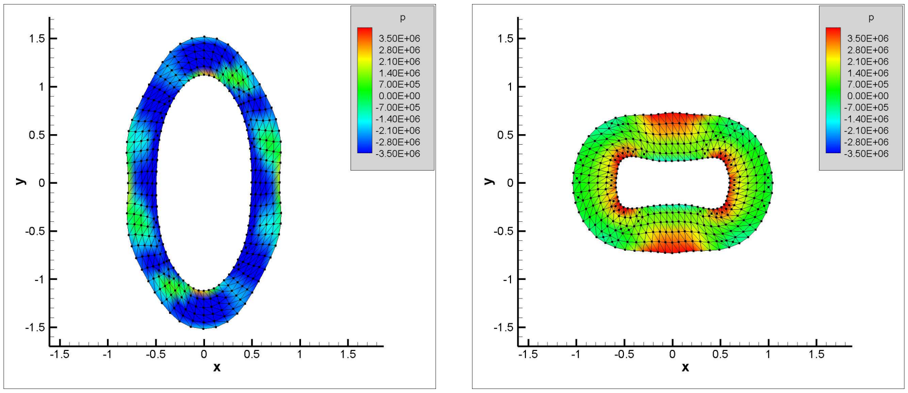

Figure 17. Example 5.2 : Ring at time $t = 0.05\,s$ (left) and $t = 0.10\,s$ (right)

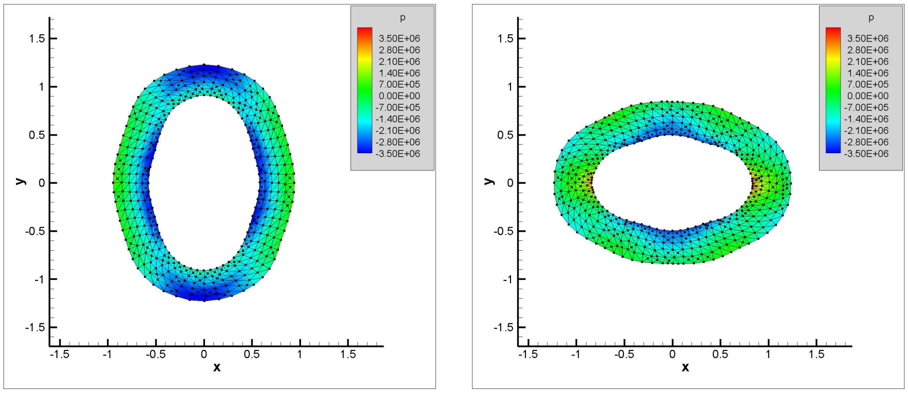

Figure 18. Example 5.2 : Ring at time $t = 0.15\,s$ (left) and $t = 0.20\,s$ (right)

remains at the same location. Viscous stabilization as described in [59], was added to reduce the error accumulation in the solution, which does not affect momentum conservation. The energy history and a close-up view are shown in figure 19. Energy spikes are observed in between the adaptation time-step pair, as shown in the close-up view. The energy at the beginning and end of the time-step pair remains well behaved.

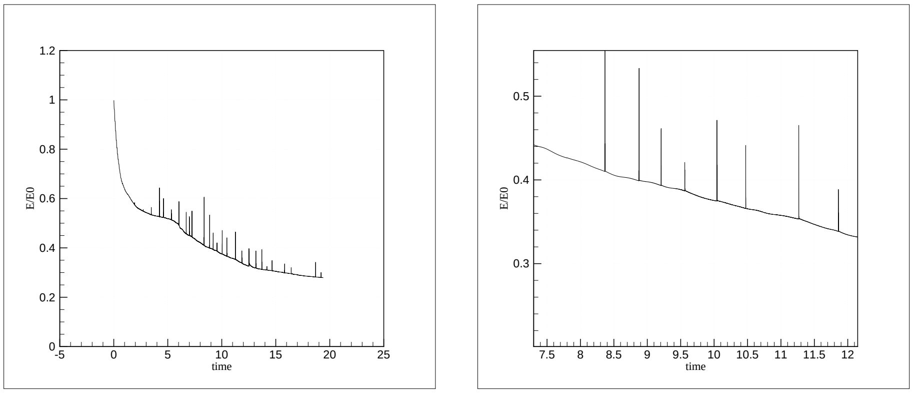

Figure 19. Example 5.2 : The figure to the left shows the total energy history of the ring. The figure to the right shows the close-up view depicting energy changes during the step

5.3. A Tensile test case

Next, a tensile test case is presented. This test case was chosen to observe the momentum behavior in presence of external forces. In order to demonstrate the exact conservation of linear and angular momentum, a modified measure of momentum is calculated.

5.3.1. Modified Momentum The measure of the modified momentum can be computed from the basic idea of measuring the momentum, in absence of external forces. Subtracting the effects of external forces from the actual momentum, the following measure is devised.

$$\boldsymbol{P}'_{n,n+1} = \boldsymbol{P}_{n,n+1} - \sum_{0}^{n} \left[\left(\int_{V_0} \rho_0 \boldsymbol{g}_n dV_0 \right) + \sum_{a \in \Gamma} \boldsymbol{R}^a_{n+1} \right] \Delta t_n \tag{139}$$

$$\boldsymbol{H}'_{n,n+1} = \boldsymbol{H}_{n,n+1} - \sum_{0}^{n} \left[\left(\int_{V_0} \rho_0 \boldsymbol{x}_n \times \boldsymbol{g}_n dV_0 \right) + \sum_{a \in \Gamma} \boldsymbol{x}_n \times \boldsymbol{R}^a_{n+1} \right] \Delta t_n \tag{140}$$

Where $\boldsymbol{g}_n$ is the external acceleration, (say gravity etc.), and is computed, like the external forces are computed, (actually external force vector could also be used), while the external nodal forces, $\boldsymbol{R}^a_{n+1}$ can be obtained while applying the boundary conditions. The modified momentum thus obtained is expected to remain conserved, in spite of presence of external forces. The conservation of modified momentum described above is not derived from the variational principle, but a means to study the momentum conservation property of the method, for problems where the total momentum is not conserved.

In the Figure 20 a square steel plate, with material properties ($E = 2.1 \times 10^{10}$ Pa, $\nu = 0.3$

and $\rho = 7 \times 10^3\,\text{kg/m}^3$) is pulled rapidly by $\boldsymbol{v}_{\text{pull}} = 40m/s$ at its top surface, and reaches thrice its length within 0.05seconds.

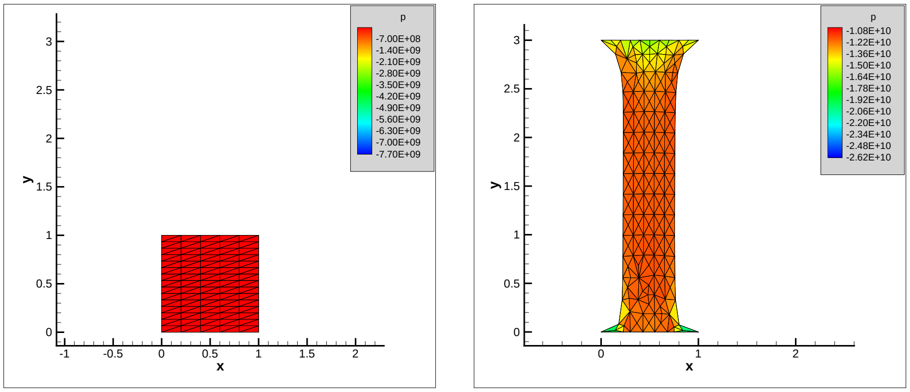

Figure 20. Example 5.3 : A Tensile test specimen (left) pulled to thrice its length (right).

Mesh adaptation was employed in the simulation, and the net momentum was conserved as is shown in Figure 21.

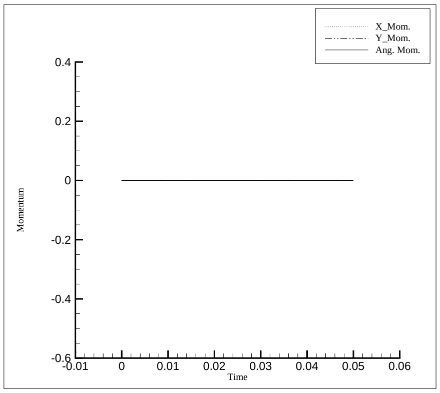

Figure 21. Example 5.3 : The Modified Momentum history for the Tensile test

5.4. A Punch test

Similar to the tensile test case another test case as that of a punching problem was considered. A flat square plate of unit length was constrained from the bottom and sides and punched into the top half with a prescribed punch velocity ($v_{\text{punch}} = 2$ m/s) as shown in Figure 22. Here a nearly incompressible rubber plate was chosen with material properties, ($E = 1.7 \times 10^7$ Pa, $\nu = 0.450$ and $\rho = 1.1 \times 10^3\,\text{kg/m}^3$). The deformed configuration at $t = 0.25$s is shown in Figure 23.

The Modified momentum remains conserved as shown in the figure 23.

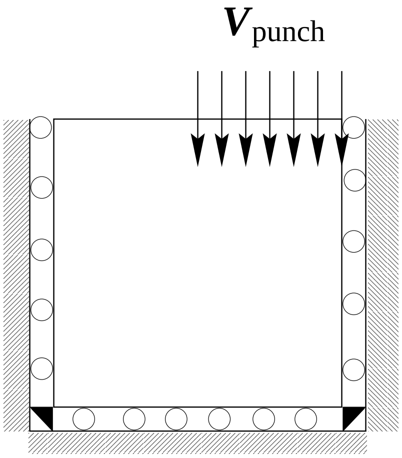

Figure 22. Example 5.4 : A schematic figure of the punch test case, showing boundary conditions.

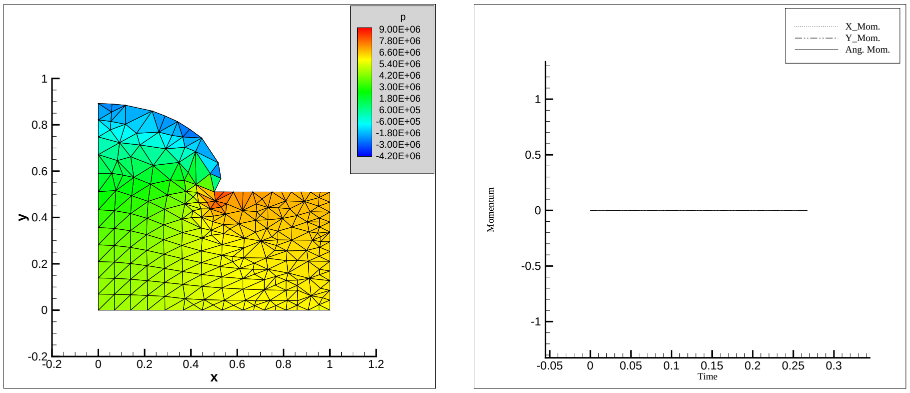

Figure 23. Example 5.4 : Mesh adaptation for a punch problem and the momentum history.

5.5. *Plate Impact*

In this example a plate impacting a rigid wall is shown. The Taylor Bar impact (TBI) is a standard benchmark in rapid dynamics problems involving large deformations. The standard TBI problem involves plastic deformations of a rod, impacting a rigid wall. In this case a plate impacting a rigid wall is considered. Since only hyperelastic materials have been considered in this paper, a modified TBI problem is presented where the constitutive relations are based on hyperelastic behavior.

In this case, a plate of length $L = 32.4$mm and width $w = 6.4$mm impacts the rigid wall with a velocity of 227m/s. Using symmetry, only half of the plate is considered with appropriate boundary conditions, as shown in figure 24. The contact boundary conditions are simply implemented by calculating the penetration of the plate node through the wall (at $y = 0$). A contact force based on the nodal mass, penetration and the time step is applied to the node:

$$
\begin{aligned}
\boldsymbol{\delta}^a_{n+1} &= \boldsymbol{x}^{wall}_{n+1} - \boldsymbol{x}^{*a}_{n+1} && (141)\\
\boldsymbol{v}^a_{n+1/2} &= \boldsymbol{v}^{*a}_{n+1/2} + \frac{\boldsymbol{\delta}^a_{n+1}}{\Delta t} && (142)\\
\boldsymbol{R}^a_{n+1/2} &= m^a_{n+1}\frac{\boldsymbol{\delta}^a_{n+1}}{\Delta t^2} && (143)\\
\boldsymbol{M}^a_{n+1/2} &= \boldsymbol{x}^a_n \times \boldsymbol{R}^a_{n+1/2} && (144)
\end{aligned}
$$

where quantities marked with asterisk are uncorrected quantities. The reaction force $\boldsymbol{R}^a_{n+1/2}$ and the reaction moment $\boldsymbol{M}^a_{n+1/2}$ are added to the net external force and external moment for calculation of the modified momentum in equations 139 and 140 respectively.

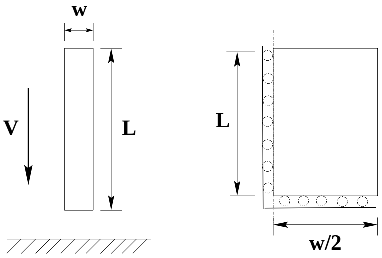

Figure 24. Example 5.5 : Schematic diagram of the plate impact problem.

The material properties of the plate were ($E = 5.85 \times 10^8$ Pa, $\nu = 0.495$, $\rho = 8930$ kg/m^3). The plate was discretized using 200 elements and the solution was computed for 194μs. Figures 25 to 28 show the solution of the deformed plate at various time instants. The mesh gets refined in the regions of high error (Z^2 error) and high mesh skewness. Figure 28 shows adaptation near the contact where the mesh undergoes the most skewness. The plate touches the wall at 14 μs. In figure 25 the plate is shown to collide with the rigid wall within the first 30 μs where the body distorts at the contact of the wall. With further motion until 60 μs, the mesh distorts

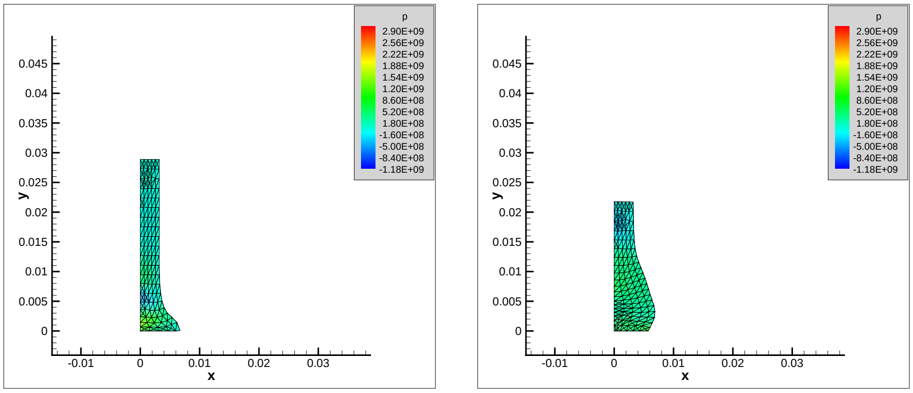

Figure 25. Example 5.5 : The plate at $t = 30\,\mu$s (left) and $t = 60\,\mu$s (right).

inside, where the mesh is adapted. All the kinetic energy of the plate is almost converted to

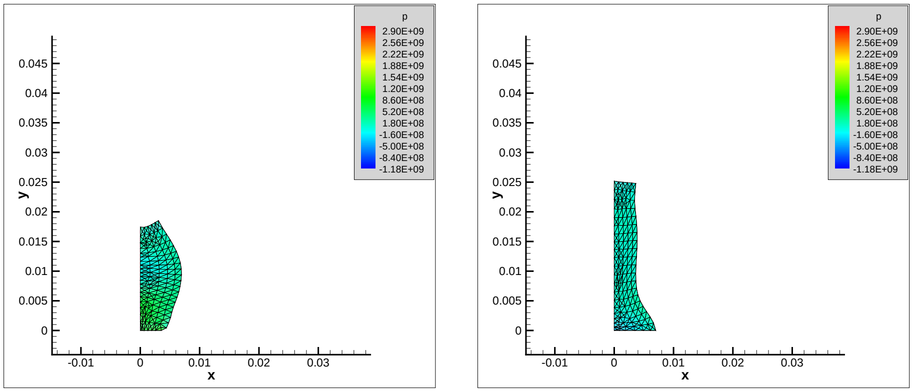

Figure 26. Example 5.5 : The plate at $t = 90\,\mu$s (left) and $t = 120\,\mu$s (right).

internal potential energy by $90\,\mu$s, almost sticking to the wall, as shown in figure 26. At $120\,\mu$s the plate begins to spring back in the opposite direction (reaction). The plate springs back until $150\,\mu$s, where it undergoes large necking type of deformation, where meshes are adapted as shown in figure 27. At roughly $180\,\mu$s, the plate leaves the rigid wall (bounce-off motion).

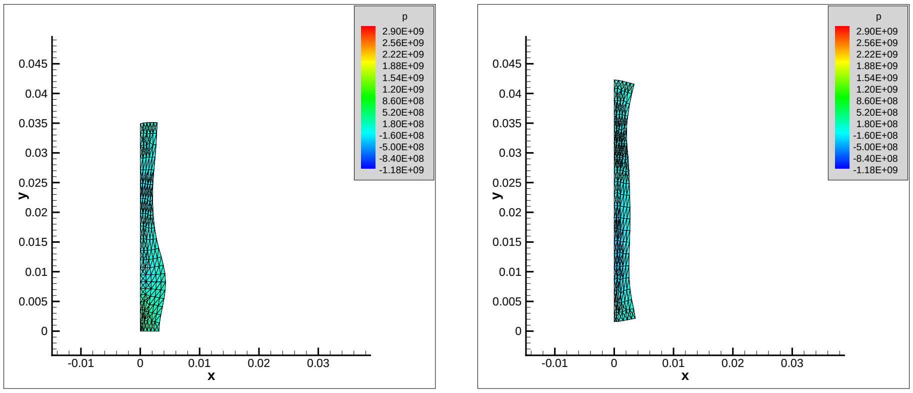

Figure 27. Example 5.5 : The plate at $t = 150\,\mu$s (left) and $t = 180\,\mu$s (right).

Figure 29, shows the velocity distribution on the deformed plate at the final time step ($194\mu s$). Due to the absence of a shock capturing numerical method, spurious modes might be present in the solution. The modified linear and angular momentum in this case remained constant throughout the computation.

Viscous stabilization as described in [59], was added to reduce the error accumulation in the solution. The total energy history as shown in figure 30. Energy spikes are observed in between the adaptation time-step pair, as shown in the close-up view. The energy at the beginning and end of the time-step pair remains well behaved.

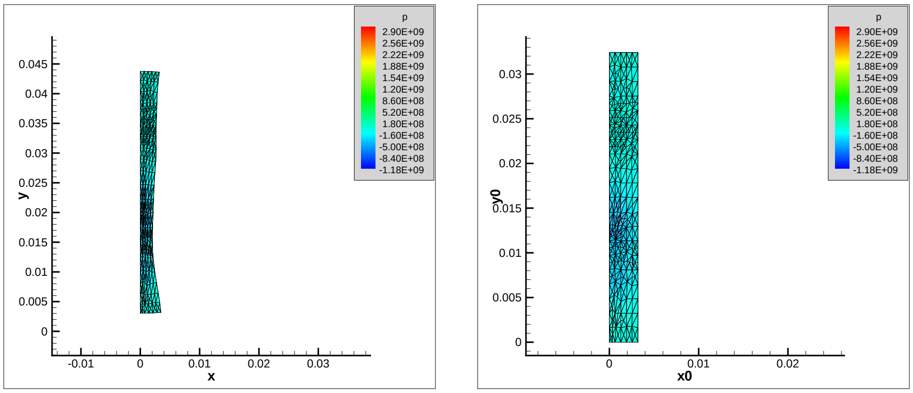

Figure 28. Example 5.5 : Deformed configuration of the plate at $t = 194\,\mu$s (left) and the corresponding mesh in the reference configuration (right).

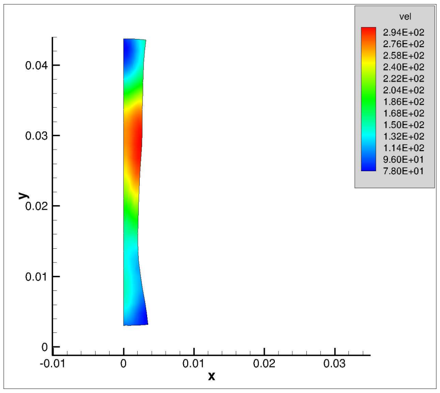

Figure 29. Example 5.5 : The velocity (m/s) distribution at $t = 194\mu s$.

6. Concluding Remarks

In this paper variationally consistent time updates for local topological changes have been developed. The methods have been formulated using the space-time discretization described in section 2. These updates have been implemented in tandem to develop a simple mesh adaptation algorithm. A simple mesh adaptive criteria based on the Z^2 error-estimate has been used. The mesh adaptation algorithm thus obtained, is shown to conserve linear and angular momentum. In cases of external forces, a modified momentum is used to demonstrate the conservation of momentum. Simple cases of rapid dynamics have been chosen to demonstrate the application of such methods. The existing adaptive procedures are explicit, and cause no significant extra expense over the standard explicit (central difference) scheme. Clearly, further work is required to augment the use of the algorithm to more complicated problems, where severe mesh distortions are encountered. Better mesh adaptation criteria would be required, to make the adaptation more effective. In those rapid dynamics problems where velocities exceed the material wave speeds, special shock capturing methods are required. Momentum conserving

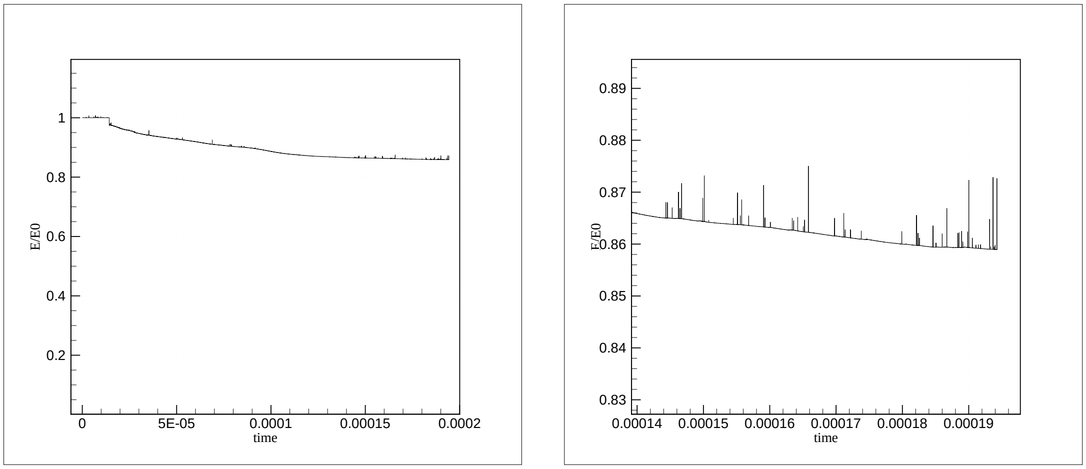

Figure 30. Example 5.5 : The figure to the left shows the total energy history of the plate. The figure to the right shows the close-up view depicting energy changes during the step

mesh-adaptive updates along with such methods could be considered in future developments.

ACKNOWLEDGEMENT

This work has been funded by Sandia National Laboratories, USA., through a collaborative research contract with MIT (Doc. No. 1152 under A0260). The authors wish to thank Dr. James R. Stewart, from Sandia National Laboratories for the research interactions and collaboration.

REFERENCES

1. Kane C., Marsden J. E., Ortiz M., and West M. Variational integrators and the newmark algorithm for conservative and dissipative mechanical systems. *Int. J. Num. Meth. Engrg.*, 49:1295–1325, 2000.
2. J. O. Hallquist. Preliminary user's manual for dyna3d and dynap. Technical Report UCID-17268, University of California, Lawrence Livermore Laboratory, Livermore, California., 1976.
3. J. O. Hallquist. Dyna2d - an explicit finite element and finite difference code for axisymmetric and plane strain calculations (user's guide). Technical Report UCRL-52429, University of California, Lawrence Livermore Laboratory, Livermore, California., March 1978.
4. W. L. Cheng, R. R. Verderber, and J. O. Hallquist. Three-dimensional simulation of forging processes. *ASME Comput. Eng. Div. Ced.*, 5:95–100, 1992.
5. S. W. Attaway, B. A. Hendrickson, S. J. Plimpton, D. R. Gardner, C. T. Vaughan, K. H. Brown, and M. W. Heinstein. A parallel contact detection algorithm for transient solid dynamics simulations using pronto3d. *Computational Mechanics*, 22:143–159, 1998.
6. S. K. Lahiri, J. Bonet, J. Peraire, and L. Casals. A variationally consistent fractional time-step integration method for incompressible and nearly incompressible lagrangian dynamics. *Int. Journal. for Numerical Methods in Engrg.*, 63:1371–1395, 2005.
7. Wendlandt J. and Marsden J. E. Mechanical integrators derived from a discrete variational principle. *Physica D*, 106:223–246, 1997.
8. Ortiz M. A note on energy conservation and stability of nonlinear time-stepping algorithms. *Computers and Structures*, 24:167–168, 1986.
9. Veselov A.P. Integrable discrete-time systems and difference operators. *Functional Analysis and its Applications*, 22:83–94, 1988.

10. A. Lew, J. E. Marsden, M. Ortiz, and M. West. Asynchronous variational integrators. *Arch. Rational. Mech. Anal.*, 167(2):85–146, April 2003.
11. Marsden J. E., Pekarsky S., and Shkoller S. Discrete euler-poincaré and lie-poisson equations. *Nonlinearity*, 12:1647–1662, 1999.
12. Gonzalez O. Time integration and discrete hamiltonian systems. *Journal of Nonlinear Science*, 6:449–469, 1996.
13. Gonzalez O. and Simo J. C. On the stability of symplectic and energy-momentum algorithms for non-linear hamiltonian systems with symmetry. *Comp. Meth. in Appl. Mech. Engrg.*, 134(3-4):197–222., 1996.
14. Cornelius Lanczos. *The Variational Principles of Mechanics*. Dover, 1986.
15. Jerrold E. Marsden and Thomas J. R. Hughes. *Mathematical Foundations of Elasticity*. Dover, 1983.
16. Simo J. C., Tarnow N., and Wong K. K. Exact energy-momentum conserving algorithms and symplectic schemes for nonlinear dynamics. *Comp. Meth. in Appl. Mech. and Engrg.*, 100:63–116, 1992.
17. Simo J. C. and Tarnow N. The discrete energy momentum integrators, conserving algorithms for nonlinear elastodynamics. *Zietschrift fuer Argewandte Mathematik und Physik*, 43:757–792, 1992.
18. Simo J. C. and Gonzalez O. Assessment of energy-momentum and symplectic schemes for stiff dynamical systems. *Proceedings of ASME Winter Annual Meeting, New Orleans*, December 1996.
19. Newmark N. M. A method of computation for structural dynamics. *ASCE Journal of Engineering Mechanics Division*, 85:64–94, 1959.
20. Pijaudier, G. Cabot, L. Bode, and A. Huerta. Arbitrary eulerian lagrangian finite element analysis of strain localization in transient problems. *Int. Journal. for Numerical Methods in Engrg.*, 38:4171–4191, 1995.
21. P. Diez and A. Huerta. A unified approach to remeshing strategies for finite element analysis. *Comput. Methods Appl. Mech. Engrg.*, 176:215–229, 1999.
22. M. Papadrakakis, G. Babilis, and P. Braouzi. Efficiency of refinement procedures for the p-version of the adaptive finite element method. *Engineering Computations*, 14(1):98–118, 1997.
23. R. Radovitzky and Ortiz M. Error estimation and adaptive meshing in strongly non-linear dynamic problems. *Comput. Methods Appl. Mech. Engrg.*, 172(1-2):203–240, 1999.
24. M. Ortiz and J. J. Quigley. Adaptive mesh refinement in strain localization problems. *Comput. Methods Appl. Mech. Engrg.*, 90:781–804, 1991.
25. M. Rivara. Local modification of meshes for adaptive and or multigrid finite element methods. *J. Comp. Appl. Math.*, 36:79–89, 1991.
26. M. Rivara. New longest edge algorithms for the refinement and/or improvement of unstructered triangulations. *Int. Journal. for Numerical Methods in Engrg.*, 40:3313–3324, 1997.
27. H. L. De Cougny and M. S. Shephard. Parallel refinement of coarsening and refinement of tetrahedral meshes. *Int. J. Numer. Meth. Engng.*, 46:1101–1125, 1999.
28. L. Freitag and P. Knupp. Tetrahedral element shape optimization via the jacobian determinant and condition number, 1999.
29. J. F. Molinari and Ortiz. M. Three-dimensional adaptive meshing by subdivision and edge-collapsing in finite-deformation dynamic-plasticity problems with application to adiabatic shear banding. *Int. J. Numer. Meth. Engng.*, 53:1101–1126, 2002.
30. G. T. Camacho and M. Ortiz. Adaptive lagrangian modeling of ballistic penetration of metallic targets. *Comput. Methods Appl. Mech. Engrg.*, 142:269–301, 1997.
31. T. D. Marusich and M. Ortiz. Modeling and simulation of high speed machining. *Int. J. Numer. Meth. Engng.*, 38:3675–3694, 1995.
32. R. C. Batra and K. I. Ko. An adaptive mesh refinement technique for the analysis of shear bands in plane strain compression of a thermoviscoplastic solid. *Computational Mechanics*, 10:369–379, 1992.
33. H. Borouchaki, P. Laug, A. Cherouat, and K. Saanouni. Adaptive remeshing in large plastic strain with damage. *Int. J. Numer. Meth. Engng.*, 63:1–36, 2005.
34. Somnath Ghosh and Suresh Raju. R-s adapted arbitrary lagrangian-eulerian finite element method for metal forming problems with strain localization. *Int. Journal. for Numerical Methods in Engrg.*, 39:3247–3272, 1996.
35. S. B. Petersen and P. A. Martins. Finite element remeshing: A metal forming approach for quadrilateral mesh generation and refinement. *Int. J. Numer. Meth. Engng.*, 40:1449–1464, 1997.
36. D. Y. Kwak, J. S. Cheon, and Y. T. Im. Remeshing for metal forming simulations-part i: Two dimensional quadrilateral remeshing. *Int. J. Numer. Meth. Engng.*, 53:2463–2500, 2002.
37. Tobias Erhart, Wolfgang Wall, and Ekkehard Ramm. Robust adaptive remeshing strategy for large deformation transient impact simulations. *Int. Journal. for Numerical Methods in Engrg.*, 65:2139–2166, 2006.
38. J. C. A. Costa and M. K. Alves. Layout optimization with *h*-adaptivity of structures. *Int. J. Numer. Meth. Engng.*, 58:83–102, 2003.
39. P. Thoutireddy and Ortiz M. A variational r-adaptation and shape-optimization method for finite-

deformation elasticity. *Int. J. Numer. Meth. Engng.*, 61:1–21, 2004.

40. I. Babuska and B. A. Szabo. On the rate of convergence of finite element method. *Int. Journal. for Numerical Methods in Engrg.*, 18:323–341, 1982.
41. D. W. Kelly, J. Gago, O. C. Zienkiewicz, and I. Babuska. A posteriori error analysis and adaptive processes in the finite element method. *Int. Journal. for Numerical Methods in Engrg.*, 19:1593–1619, 1983.
42. O. C. Zienkiewicz and J. Z. Zhu. A simple error estimator and adaptive procedure for practical engineering analysis. *Int. Journal. for Numerical Methods in Engrg.*, 24:337–357, 1987.
43. X. D. Li and N. E. Wiberg. A postprocessed error estimate and an adaptive procedure for the semi-discrete finite element in dynamic analysis. *Int. J. Numer. Meth. Engng.*, 37:3585–3603, 1994.
44. G. M. Hulbert and I. Jang. Automatic time-step control algorithms for structural dynamics. *Comput. Methods Appl. Mech. Engrg.*, 126:155–178, 1995.
45. Thomas Grätsch and Klaus-Jürgen Bathe. A posteriori error estimation techniques in practical finite element analysis. *Computers and Structures*, 83:235–265, 2005.
46. O. C. Zienkiewicz and J. Z. Zhu. The superconvergent patch recovery and a-posteriori error estimates, part 1: The recovery technique. *Int. Journal. for Numerical Methods in Engrg.*, 33:1331–1364, 1992.
47. O. C. Zienkiewicz and J. Z. Zhu. The superconvergent patch recovery and a-posteriori error estimates, part 2: Error estimates and adaptivity. *Int. Journal. for Numerical Methods in Engrg.*, 33:1365–1382, 1992.
48. Pierre Ladevéze and Nicolas Moës. A new *a posteriori* error estimation for nonlinear time-dependant finite element analysis. *Comput. Methods Appl. Mech. Engrg.*, 157:45–68, 1997.
49. J. P. Combe, Pierre Ladevéze, and J. P. Pelle. Discretization error estimation for transient dynamic simulation. *Advances in Engineering Software*, 33:553–563, 2002.
50. Pierre Ladevéze and J. P. Pelle. *Mastering Calculations in Linear and Nonlinear Mechanics.* Springer, Mechanical Engineering Series, 2004.
51. Weizhang Huang. Variational mesh adaptation: isotropy and equidistribution. *J. Comput. Phys.*, 174(2):903–924, 2001.
52. Weizhang Huang and Weiwei Sun. Variational mesh adaptation ii: error estimates and monitor functions. *J. Comput. Phys.*, 184(2):619–648, 2003.
53. W. Cao, R. C. Gonzalez, W. Huang, and R. D. Russell. Variational mesh adaptation methods for axisymmetrical problems. *SIAM J. Numer. Anal.*, 41(1):235–257, 2003.
54. M. G. Zielonka, Ortiz M., and J. E. Marsden. Variational r-adaptation in elastodynamics. *Int. J. Numer. Meth. Engng.*, 74:1162–1197, 2008.
55. Gurtin, M.E. *Configurational Forces as Basic Concepts of Continuum Physics.* Springer, New York, 2000.
56. E. Stein (ed.). *Error-Controlled Adaptive FEM in Solid Mechanics.* Wiley, 2003.
57. Javier Bonet and Richard D. Wood. *Nonlinear continuum mechanics for finite element analysis.* Cambridge University Press, 1997.
58. Gerhard A. Holzapfel. *Nonlinear Solid Mechanics.* John Wiley & Sons Ltd., 2000.
59. Sudeep K. Lahiri. *Variationally consistent methods for Lagrangian dynamics for continuum mechanics.* PhD thesis, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, June 2006. (http://dspace.mit.edu/handle/1721.1/39597).