INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING
Int. J. Numer. Meth. Engng 2007; **00**:1–38 *Prepared using* nmeauth.cls *[Version: 2002/09/18 v2.02]*

A Variationally Consistent Mesh Adaptation Method for 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, MA, USA.*
[2] *Civil & Computational Engineering Center, School of Engineering, University of Wales, Swansea, 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

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 numerical problems lie in the proper modeling of large deformations and rotations, contact, and

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

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

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 do not ensure conservation of momentum which 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. The difference of the reference and the elemental stresses, provides for an error estimate. This type of error indicator can be classified under gradient based errors indicators. Curvature based error-estimates have been used by [33], in problems of large plastic strain damage. Another approach has been found in [24, 32, 30] where gradients in direct physical quantities like the velocity field or strain field or other choice of quantities are used as empirical adaptation criteria. Recently, [48, 49, 50] have developed a new approach for error-estimation based on the constitutive relation error. They describe the finite element solution as a displacement-stress pair $(\hat{\boldsymbol{U}}_h, \hat{\boldsymbol{\sigma}}_h)$ such that the displacements satisfy kinematic constraints like boundary conditions and initial conditions while the stresses satisfy the equilibrium conditions. The displacements and stresses do not satisfy the constitutive relations (stress-strain relations) which provides an error measure which they refer to as the constitutive relation error. This error measure has been found to be effective in large strain transient problems. Similarly error-estimators based on the time update (using semi-discrete equations of motion) are formulated in [51]. 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, 52, 53, 54]. Recently, some researchers [39], have used the idea of configurational forces [55] for r-adaptation, for applications in shape optimization. Configurational forces are obtained as a variation of the internal energy with respect to material position vectors. This leads to the criteria to move mesh points to obtain an optimal mesh, which also leads to shape

optimization. An overview of various error-estimation techniques and adaptation criteria can be found in [45, 56].

Significant research has been done in the fluid mechanics community in the area of mesh adaptation, where the main emphasis has been in proper resolution of the flow field, especially in the simulation of boundary layers, shock waves and high speed compressible flows. Several researchers like [57, 58, 59, 60, 61, 62, 63] have developed very effective mesh adaptive solvers for compressible flows. Most of these adaptations are based on error-estimates which are based on gradients of flow properties. Use of error estimators based on bounds on functional outputs [64, 65] have also proven to be very effective in the calculation of important aerodynamic properties like lift or drag of an airfoil in presence of shocks and viscous effects.

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 $\rho(t)$.

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

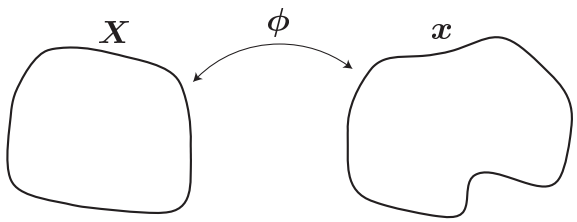

Figure 1. Continuous systems

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

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

The above expression is well suited for compressible or nearly incompressible materials, [66, 67].

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

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

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, that is,

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

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, 68]. The discrete Lagrangian Integral can be further split into the Kinetic Energy Integral and the Potential Energy Integrals as:

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

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 :

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

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:

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

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:

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

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:

$$\boldsymbol{x}_e^n \quad = \quad N_a^e \boldsymbol{x}_a^n \tag{13}$$

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

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

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:

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

The stationarity condition then becomes:

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

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

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

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 $\boldsymbol{x}_a^n$ 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 $\boldsymbol{x}_a^n$ 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:

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

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:

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

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

where

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

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

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

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

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, (from connectivity) 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:

$$\begin{aligned} \frac{\partial \Pi^{\text{int}}(\boldsymbol{x})}{\partial x_i^b} &= \sum_{(e,a)\in b} \int_{V_e} \frac{\partial N_a^e}{\partial x_j} \sigma_{ij}^e dV_e \\ &= T_i^b = \sum_{(e,a)\in b} T_{ai}^e \end{aligned} \tag{28}$$

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 [68] 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 mass. 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 ($\boldsymbol{x}_b^n$) is:

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

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. We begin with a single triangular element.

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:

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

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

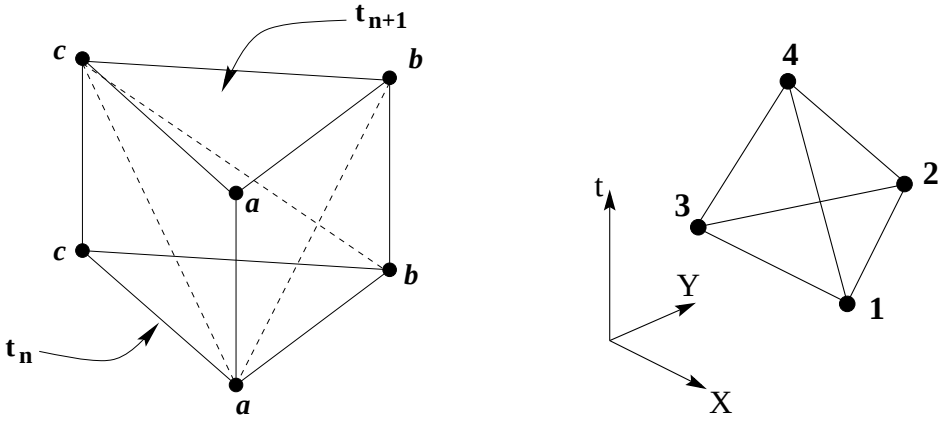

Figure 2. The space-time-prism (left) and a generic space-time-tetrahedron (right).

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:

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

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

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

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

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

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 criteria 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:

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

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 simplified 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:

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

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

$$K_{n,n+1}(\boldsymbol{x}_n, \boldsymbol{x}_{n+1}) \quad = \quad \sum_a \frac{\Delta t}{2} M^a \; \boldsymbol{v}^a_{n+1/2} \cdot \boldsymbol{v}^a_{n+1/2} \tag{45}$$

Where M^a is the lumped mass of each node(a) in the mesh. Thus, the net Lagrangian of the entire mesh becomes:

$$L_{n,n+1}(\boldsymbol{x}_n, \boldsymbol{x}_{n+1}) \quad = \quad \sum_a \frac{\Delta t}{2} M^a \; \boldsymbol{v}^a_{n+1/2} \cdot \boldsymbol{v}^a_{n+1/2} \quad - \quad \Delta t \Pi(\boldsymbol{x}_n) \tag{46}$$

This leads to the discrete Lagrangian Integral of the Central Difference method, as discussed in [68, 6].

So far in this section some generic space-time discretization principles have been presented and used to develop the standard Central Difference method. It was shown that the Central Difference method, with lumped mass, can be interpreted as an outcome of linear interpolations in space-time within a variational framework. The objective was to present the generic treatment of space-time discretization, which would be extended in the later sections for cases where the mesh topology changes over time. Now the Kinetic Energy Integral is revisited to evaluate the directional derivatives:

$$K_{n,n+1}(\boldsymbol{x}_n, \boldsymbol{x}_{n+1}) \quad = \quad \sum_a \frac{\Delta t}{2} M^a \boldsymbol{v}^a_{n+1/2} \cdot \boldsymbol{v}^a_{n+1/2} \tag{47}$$

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

$$\begin{aligned} \boldsymbol{v}^a_{n+1/2} \quad &= \quad \frac{1}{\Delta t} \left(\boldsymbol{x}^a_{n+1} - \boldsymbol{x}^a_n \right) \\ \frac{\partial K_{n,n+1}}{\partial \boldsymbol{x}^n_b} \quad &= \quad -M^b \boldsymbol{v}^b_{n+1/2} \end{aligned} \tag{48}$$

Similarly :

$$\frac{\partial K_{n-1,n}}{\partial \boldsymbol{x}_b^n} = M^b \boldsymbol{v}_{n-1/2}^b \tag{49}$$

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

$$\frac{\partial K_{n,n+1}}{\partial \boldsymbol{x}_b^n} - \Delta t \frac{\partial \Pi_n^{\text{ext}}}{\partial \boldsymbol{x}_b^n} - \Delta t \frac{\partial \Pi_n^{\text{int}}}{\partial \boldsymbol{x}_b^n} + \frac{\partial K_{n-1,n}}{\partial \boldsymbol{x}_b^n} = M^b \left(-\boldsymbol{v}_{n+1/2}^b + \boldsymbol{v}_{n-1/2}^b \right) - \Delta t \left(\boldsymbol{T}_n^b - \boldsymbol{F}_n^b \right) = 0$$

which can be written in the vector form:

$$M^b \left(\boldsymbol{v}_{n+1/2}^b - \boldsymbol{v}_{n-1/2}^b \right) = \Delta t \left(\boldsymbol{F}_n^b - \boldsymbol{T}_n^b \right) \tag{50}$$

Thus, we obtain the time marching algorithm with the Central Difference Scheme, using the spatial discretization of the standard linear element. The standard linear element described above is commonly found in literature [69]. The purpose of deriving the commonly known central difference method, was to demonstrate that it belongs to the class of variational integrators, [1]. 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.

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

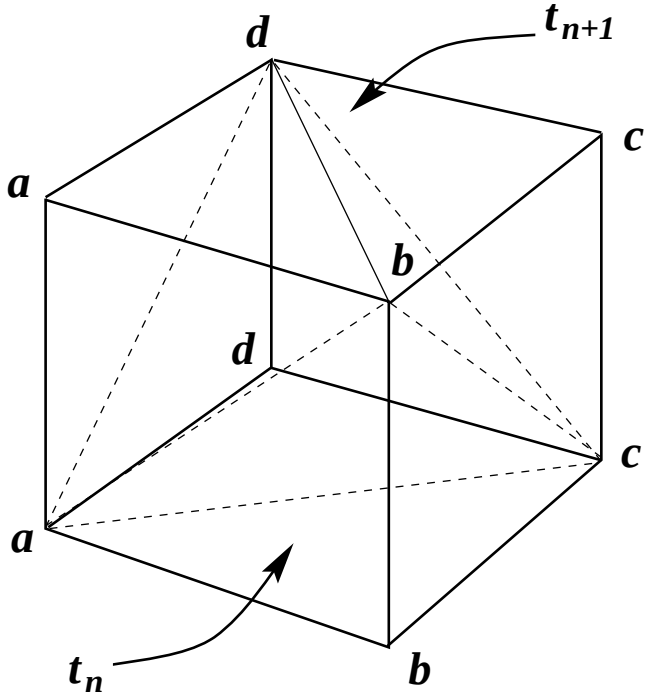

Figure 3. The space-time volume for the diagonal swapping.

as:

$$K^{abcd}_{n,n+1} = \frac{\Delta t}{2}\frac{m_{abc}}{3}\boldsymbol{v}^b_{n+1/2}\cdot\boldsymbol{v}^b_{n+1/2} + \frac{\Delta t}{2}\frac{m_{acd}}{3}\boldsymbol{v}^d_{n+1/2}\cdot\boldsymbol{v}^d_{n+1/2} + \frac{\Delta t}{2}\frac{m_{abd}}{3}\boldsymbol{v}^a_{n+1/2}\cdot\boldsymbol{v}^a_{n+1/2} + \frac{\Delta t}{2}\frac{m_{bcd}}{3}\boldsymbol{v}^c_{n+1/2}\cdot\boldsymbol{v}^c_{n+1/2} + \frac{\Delta t}{2}\frac{m_{abcd}}{3}\boldsymbol{v}^{abcd}_{n+1/2}\cdot\boldsymbol{v}^{abcd}_{n+1/2} \quad (51)$$

$$\boldsymbol{v}^{abcd}_{n+1/2} = \frac{\left(m_{acd}\boldsymbol{x}^b_{n+1} + m_{abc}\boldsymbol{x}^d_{n+1} - m_{bcd}\boldsymbol{x}^a_n + m_{abd}\boldsymbol{x}^c_n\right)}{\Delta t m_{abcd}} \quad (52)$$

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:

$$\begin{aligned} -D_1K^{abcd}_{n,n+1}[\delta\boldsymbol{x}_n] &= \frac{m_{abc}}{3}\boldsymbol{v}^b_{n+1/2}\cdot\delta\boldsymbol{x}^b_n + \frac{m_{acd}}{3}\boldsymbol{v}^d_{n+1/2}\cdot\delta\boldsymbol{x}^d_n \\ &+ \left(\frac{m_{abd}}{3}\boldsymbol{v}^a_{n+1/2} + \frac{m_{bcd}}{3}\boldsymbol{v}^{abcd}_{n+1/2}\right)\cdot\delta\boldsymbol{x}^a_n \\ &+ \left(\frac{m_{bcd}}{3}\boldsymbol{v}^c_{n+1/2} + \frac{m_{abd}}{3}\boldsymbol{v}^{abcd}_{n+1/2}\right)\cdot\delta\boldsymbol{x}^c_n \end{aligned} \quad (53)$$

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:

$$M^n_b\left(\boldsymbol{v}^{n+1/2}_b - \boldsymbol{v}^{n-1/2}_b\right) = \Delta t\left(\boldsymbol{F}^b_n - \boldsymbol{T}^b_n\right) \quad (54)$$

$$M^n_d\left(\boldsymbol{v}^{n+1/2}_d - \boldsymbol{v}^{n-1/2}_d\right) = \Delta t\left(\boldsymbol{F}^d_n - \boldsymbol{T}^d_n\right) \quad (55)$$

Note that as soon as the position of nodes b and d have been updated, using equations 54 and 55, it is possible to calculate $\boldsymbol{v}_{n+1/2}^{abcd}$ using 52 which in turns allows the update of a and c to take place as:

$$M_a^{n+1}\boldsymbol{v}_a^{n+1/2} - M_a^n\boldsymbol{v}_a^{n-1/2} + \frac{m_{bcd}}{3}\boldsymbol{v}_{abcd}^{n+1/2} = \Delta t\left(\boldsymbol{F}_n^a - \boldsymbol{T}_n^a\right) \tag{56}$$

$$M_c^{n+1}\boldsymbol{v}_c^{n++/2} - M_c^n\boldsymbol{v}_c^{n-1/2} + \frac{m_{abd}}{3}\boldsymbol{v}_{abcd}^{n+1/2} = \Delta t\left(\boldsymbol{F}_n^c - \boldsymbol{T}_n^c\right) \tag{57}$$

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

$$\begin{aligned} -D_2K_{n,n+1}^{abcd}[\delta\boldsymbol{x}_{n+1}] &= \frac{m_{abd}}{3}\boldsymbol{v}_{n+1/2}^a\cdot\delta\boldsymbol{x}_{n+1}^a + \frac{m_{bcd}}{3}\boldsymbol{v}_{n+1/2}^c\cdot\delta\boldsymbol{x}_{n+1}^c \\ &+ \left(\frac{m_{abc}}{3}\boldsymbol{v}_{n+1/2}^b + \frac{m_{acd}}{3}\boldsymbol{v}_{n+1/2}^{abcd}\right)\cdot\delta\boldsymbol{x}_{n+1}^b \\ &+ \left(\frac{m_{acd}}{3}\boldsymbol{v}_{n+1/2}^d + \frac{m_{abc}}{3}\boldsymbol{v}_{n+1/2}^{abcd}\right)\cdot\delta\boldsymbol{x}_{n+1}^d \end{aligned} \tag{58}$$

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

$$M_a^{n+1}\left(\boldsymbol{v}_a^{n+3/2} - \boldsymbol{v}_a^{n+1/2}\right) = \Delta t\left(\boldsymbol{F}_{n+1}^a - \boldsymbol{T}_{n+1}^a\right) \tag{59}$$

$$M_c^{n+1}\left(\boldsymbol{v}_c^{n+3/2} - \boldsymbol{v}_c^{n+1/2}\right) = \Delta t\left(\boldsymbol{F}_{n+1}^c - \boldsymbol{T}_{n+1}^c\right) \tag{60}$$

$$M_b^{n+1}\boldsymbol{v}_b^{n+3/2} - M_b^n\boldsymbol{v}_b^{n+1/2} - \frac{m_{acd}}{3}\boldsymbol{v}_{abcd}^{n+1/2} = \Delta t\left(\boldsymbol{F}_{n+1}^b - \boldsymbol{T}_{n+1}^b\right) \tag{61}$$

$$M_d^{n+1}\boldsymbol{v}_d^{n+3/2} - M_d^n\boldsymbol{v}_d^{n+1/2} - \frac{m_{abc}}{3}\boldsymbol{v}_{abcd}^{n+1/2} = \Delta t\left(\boldsymbol{F}_{n+1}^d - \boldsymbol{T}_{n+1}^d\right) \tag{62}$$

The momentum within time step t_n and t_{n+1} is $(D_2L_{n,n+1})$:

$$\boldsymbol{P}_{n,n+1} = \sum_j \boldsymbol{P}_{n,n+1}^j \tag{63}$$

$$\boldsymbol{P}_{n,n+1}^j = \begin{cases} M_j^{n+1}\boldsymbol{v}_j^{n+1/2} & \text{for } j = a \text{ or } c, \\ M_j^n\boldsymbol{v}_j^{n+1/2} + (\frac{m_{acd}}{3})\boldsymbol{v}_{n+1/2}^{abcd} & \text{for } j = b, \\ M_j^n\boldsymbol{v}_j^{n+1/2} + (\frac{m_{abc}}{3})\boldsymbol{v}_{n+1/2}^{abcd} & \text{for } j = d. \end{cases} \tag{64}$$

$$\boldsymbol{H}_{n,n+1} = \sum_j \boldsymbol{x}_{n+1}^j \times \boldsymbol{P}_{n,n+1}^j \tag{65}$$

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: $(a_nb_nd_na_{n+1})$, $(b_nc_nd_nc_{n+1})$, $(a_nc_nd_nd_{n+1})$, $(a_nc_nb_nb_{n+1})$, and $(b_nd_na_{n+1}c_{n+1})$. 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

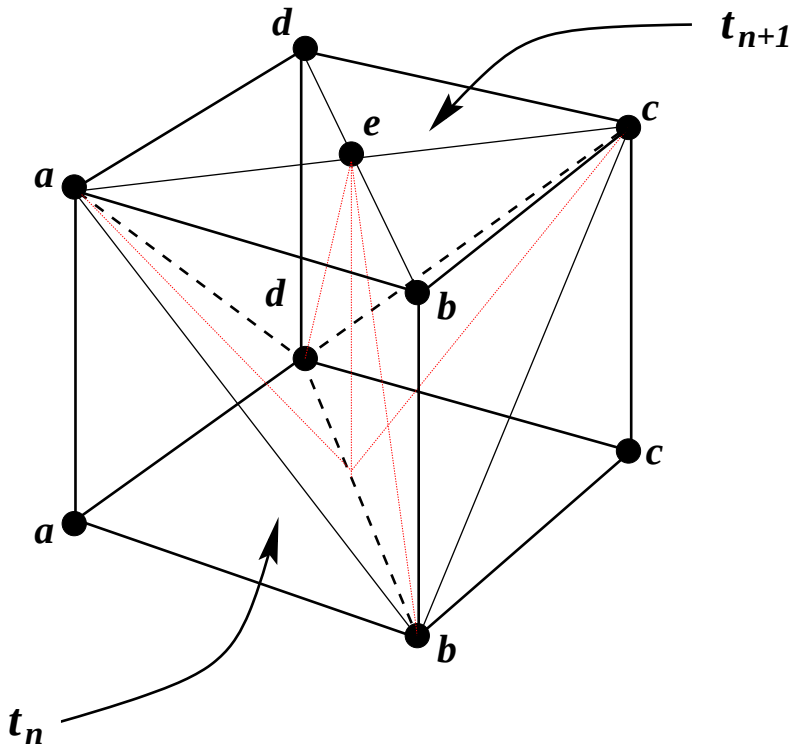

Figure 4. The space-time volume for edge splitting.

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 Energy Integral can be written as:

$$
\begin{aligned}
K^{abcd}_{n,n+1} &= \frac{\Delta t}{2}\frac{m_{abd}}{3}\boldsymbol{v}^a_{n+1/2}\cdot\boldsymbol{v}^a_{n+1/2} + \frac{\Delta t}{2}\frac{m_{bcd}}{3}\boldsymbol{v}^c_{n+1/2}\cdot\boldsymbol{v}^c_{n+1/2} \\
&+ \frac{\Delta t}{2}\frac{m_{abe}+m_{bce}}{3}\boldsymbol{v}^c_{n+1/2}\cdot\boldsymbol{v}^c_{n+1/2} + \frac{\Delta t}{2}\frac{m_{dae}+m_{dec}}{3}\boldsymbol{v}^c_{n+1/2}\cdot\boldsymbol{v}^c_{n+1/2} \\
&+ \frac{\Delta t}{2}m_e\boldsymbol{v}^e_{n+1/2}\cdot\boldsymbol{v}^e_{n+1/2} \qquad (66)\\
m_e &= \frac{(m_{abd}+m_{bcd})}{3} \qquad (67)\\
\boldsymbol{v}^e_{n+1/2} &= \frac{1}{\Delta t}\left[\boldsymbol{x}^e_{n+1} - \frac{\left(\boldsymbol{x}^b_n + \boldsymbol{x}^d_n\right)}{2}\right] \qquad (68)
\end{aligned}
$$

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

$$\begin{aligned} -D_1 K^{abcd}_{n,n+1}[\delta \boldsymbol{x}_n] &= \frac{m_{abd}}{3}\boldsymbol{v}^a_{n+1/2}\cdot\delta\boldsymbol{x}^a_n + \frac{m_{bcd}}{3}\boldsymbol{v}^c_{n+1/2}\cdot\delta\boldsymbol{x}^c_n \\ &+ \frac{m_{abe}+m_{bce}}{3}\boldsymbol{v}^b_{n+1/2}\cdot\delta\boldsymbol{x}^b_n + \frac{m_{dae}+m_{dec}}{3}\boldsymbol{v}^d_{n+1/2}\cdot\delta\boldsymbol{x}^d_n \\ &+ \frac{m_e}{2}\boldsymbol{v}^e_{n+1/2}\cdot\delta\boldsymbol{x}^b_n + \frac{m_e}{2}\boldsymbol{v}^e_{n+1/2}\cdot\delta\boldsymbol{x}^d_n \end{aligned} \tag{69}$$

Thus, the update algorithm at step t_n :

$$M^n_a\left(\boldsymbol{v}^{n+1/2}_a - \boldsymbol{v}^{n-1/2}_a\right) = \Delta t\left(\boldsymbol{F}^a_n - \boldsymbol{T}^a_n\right) \tag{70}$$

$$M^n_c\left(\boldsymbol{v}^{n+1/2}_c - \boldsymbol{v}^{n-1/2}_c\right) = \Delta t\left(\boldsymbol{F}^c_n - \boldsymbol{T}^c_n\right) \tag{71}$$

$$M^{n+1}_b\boldsymbol{v}^{n+1/2}_b - M^n_b\boldsymbol{v}^{n-1/2}_b - \frac{m_e}{2}\boldsymbol{v}^{n+1/2}_e = \Delta t\left(\boldsymbol{F}^a_n - \boldsymbol{T}^a_n\right) \tag{72}$$

$$M^{n+1}_d\boldsymbol{v}^{n+1/2}_d - M^n_d\boldsymbol{v}^{n-1/2}_d - \frac{m_e}{2}\boldsymbol{v}^{n+1/2}_e = \Delta t\left(\boldsymbol{F}^d_n - \boldsymbol{T}^d_n\right) \tag{73}$$

Choosing the velocity at the new node to be:

$$\boldsymbol{v}^{n+1/2}_e = \frac{1}{2}\left(\boldsymbol{v}^{n+1/2}_b + \boldsymbol{v}^{n+1/2}_d\right) \tag{74}$$

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

$$\begin{bmatrix} M^{n+1}_b + \frac{1}{4}m_e & \frac{1}{4}m_e \\ \frac{1}{4}m_e & M^{n+1}_d + \frac{1}{4}m_e \end{bmatrix}\begin{bmatrix} \boldsymbol{v}^{n+1/2}_b \\ \boldsymbol{v}^{n+1/2}_d \end{bmatrix} = \begin{bmatrix} M^n_b\boldsymbol{v}^{n+1/2}_b \\ M^n_d\boldsymbol{v}^{n+1/2}_d \end{bmatrix} + \Delta t\begin{bmatrix} \boldsymbol{F}^b_n - \boldsymbol{T}^b_n \\ \boldsymbol{F}^d_n - \boldsymbol{T}^d_n \end{bmatrix} \tag{75}$$

The update at equations at $n+1$ are unchanged. The momentum within time step t_n and t_{n+1} is $(D_2 L_{n,n+1})$:

$$\boldsymbol{P}_{n,n+1} = \sum_j \boldsymbol{P}^j_{n,n+1} \tag{76}$$

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

$$\boldsymbol{H}_{n,n+1} = \sum_j \boldsymbol{x}^j_{n+1}\times\boldsymbol{P}^j_{n,n+1} \tag{77}$$

3.3. Node Movement

In order to initiate a study of 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:

$$\boldsymbol{v} = \frac{\partial}{\partial t}\phi(\boldsymbol{X},t); \qquad \nu = \frac{\partial}{\partial t}\varphi(\xi,t); \qquad \boldsymbol{V} = \frac{\partial}{\partial t}\psi(\xi,t); \tag{78}$$

$$\boldsymbol{v} = \nu - \boldsymbol{F}\boldsymbol{V} \tag{79}$$

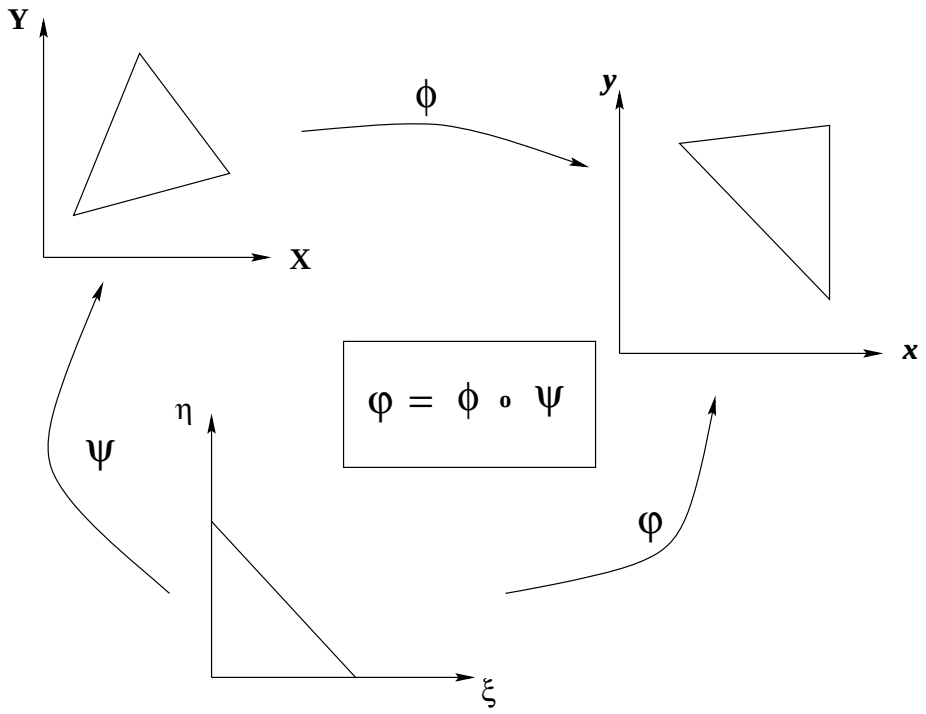

Figure 5. Understanding node movement with an intermediate mapping.

The Kinetic Energy can then be written as:

$$K_{n,n+1}\left(\boldsymbol{x}_n, \boldsymbol{x}_{n+1}\right) = \frac{\Delta t}{2} \sum_I M_I^{n+1/2} \boldsymbol{v}_{n,n+1}^I \cdot \boldsymbol{v}_{n,n+1}^I \tag{80}$$

where:

$$M_I^{n+1/2} = \frac{1}{2}\left(M_I^n + M_I^{n+1}\right) \tag{81}$$

$$\boldsymbol{v}_{n,n+1}^I = \nu_{n+1/2}^I - \boldsymbol{F}_n^I \boldsymbol{V}_{n+1/2}^I \tag{82}$$

$$\nu_{n+1/2}^I = \frac{1}{\Delta t}\left(\boldsymbol{x}_{n+1}^I - \boldsymbol{x}_n^I\right) \tag{83}$$

$$\boldsymbol{V}_{n+1/2}^I = \frac{1}{\Delta t}\left(\boldsymbol{X}_{n+1}^I - \boldsymbol{X}_n^I\right) \tag{84}$$

$$\boldsymbol{F}_n^I = \left(\frac{\int_{V_0} N_I^n \boldsymbol{F}_n dm}{\int_{V_0} N_I^n dm}\right) \tag{85}$$

The deformation gradient $\boldsymbol{F}_n$ as used in equation 82 is evaluated at time level n in order to make the update explicit. The corresponding equilibrium equations are (for any generic node I, and its neighboring nodes J):

$$M_I^{n+1/2} \boldsymbol{v}_{n,n+1}^I - M_I^{n-1/2} \boldsymbol{v}_{n-1,n}^I = -\Delta t \boldsymbol{Q}_{n,n+1}^I + \Delta t\left(\boldsymbol{F}_n^I - \boldsymbol{T}_n^I\right) \tag{86}$$

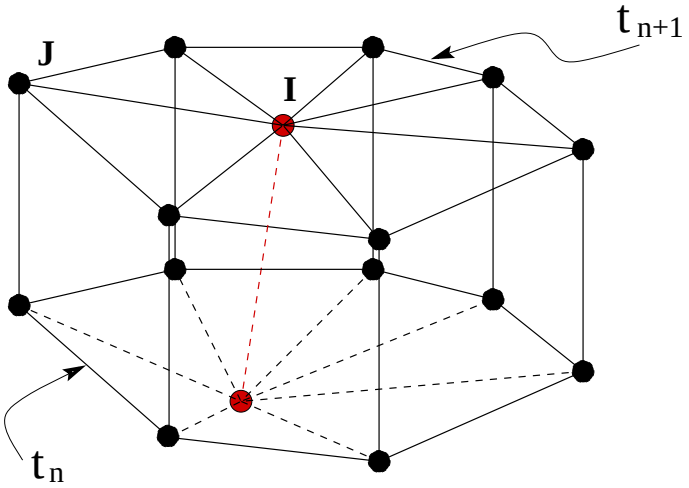

Figure 6. Space-time volume for node movement.

where:

$$\boldsymbol{v}^I_{n,n+1} = \frac{1}{\Delta t}\left[\left(\boldsymbol{x}^I_{n+1} - \boldsymbol{x}^I_n\right) - \boldsymbol{F}^I_n\left(\boldsymbol{X}^I_{n+1} - \boldsymbol{X}^I_n\right)\right] \tag{87}$$

$$\boldsymbol{Q}^I_{n,n+1} = \sum_J \frac{m^{n+1/2}_J}{m^n_J}\left(\boldsymbol{v}^J_{n,n+1} \otimes \boldsymbol{V}^J_{n,n+1}\right)\int_{V_0} N^n_J \nabla_0 N^n_I dm \tag{88}$$

This expression remains explicit if the neighboring J nodes remain fixed i.e. ($\boldsymbol{V}^J_{n,n+1} = 0$) in which case:

$$\boldsymbol{Q}^I_{n,n+1} = \left(\boldsymbol{v}^I_{n,n+1} \otimes \boldsymbol{V}^I_{n,n+1}\right)\int_{V_0} N^n_I \nabla_0 N^n_I dm = 0$$

Provided that either I is an internal node or $\boldsymbol{V}^I$ remains tangential to the reference boundary Γ_0. Thus for the node to be moved (I) the update step becomes:

$$M^{n+1/2}_I \boldsymbol{v}^I_{n,n+1} - M^{n-1/2}_I \boldsymbol{v}^I_{n-1,n} = \Delta t\left(\boldsymbol{F}^I_n - \boldsymbol{T}^I_n\right) \tag{89}$$

And for the neighboring nodes (J) the update step becomes:

$$M^{n+1/2}_J \boldsymbol{v}^J_{n,n+1} - M^{n-1/2}_J \boldsymbol{v}^J_{n-1,n} = -\Delta t\,\boldsymbol{Q}^J_{n,n+1} + \Delta t\left(\boldsymbol{F}^J_n - \boldsymbol{T}^J_n\right) \tag{90}$$

$$\boldsymbol{Q}^J_{n,n+1} = \left(\boldsymbol{v}^I_{n,n+1} \otimes \boldsymbol{V}^I_{n,n+1}\right)\int_{V_0} N^n_I \nabla_0 N^n_J dm \tag{91}$$

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

$$\boldsymbol{P}_{n,n+1} = \sum_j \boldsymbol{P}^j_{n,n+1} \tag{92}$$

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

$$\boldsymbol{H}_{n,n+1} = \sum_j \boldsymbol{x}^j_{n+1} \times \boldsymbol{P}^j_{n,n+1} \tag{93}$$

3.4. Edge Collapsing

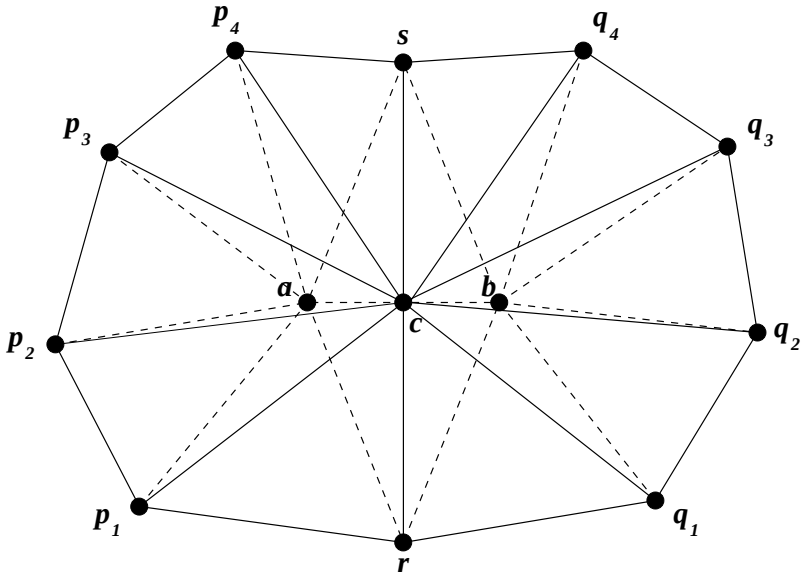

Figure 7. Collapsing the edge ab to the point c.

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, are labeled type (II) and the ones having c as one of their vertices 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.

$$\boldsymbol{X}^c_{n+1} = \xi \boldsymbol{X}^a_n + \eta \boldsymbol{X}^b_n + (1-\xi-\eta)\boldsymbol{X}^r_n \tag{94}$$

$$\boldsymbol{x}^c_n = \xi \boldsymbol{x}^a_n + \eta \boldsymbol{x}^b_n + (1-\xi-\eta)\boldsymbol{x}^r_n \tag{95}$$

The Kinetic Energy Integral and the velocity interpolation within the tetrahedra of type (I) can be written as follows:

$$K^I_{n,n+1} = \frac{\Delta t}{2}\frac{m^n_{ab}}{3}\boldsymbol{v}^c_{n+1/2}\cdot\boldsymbol{v}^c_{n+1/2} \tag{96}$$

$$m^n_{ab} = m^n_{arb} + m^n_{abs}$$

$$v^c_{n+1/2} = \frac{\boldsymbol{x}^c_{n+1} - (\xi\boldsymbol{x}^a_n + \eta\boldsymbol{x}^b_n + (1-\xi-\eta)\boldsymbol{x}^r_n)}{\Delta t} \tag{97}$$

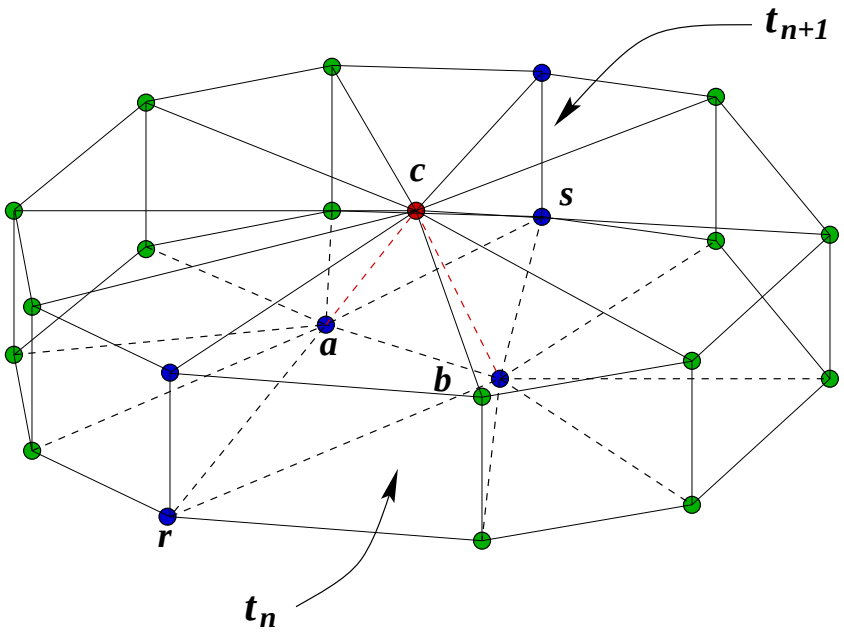

Figure 8. The space-time volume for Edge collapsing operation.

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

$$K^{II}_{n,n+1} = \frac{\Delta t}{2} m^n_{g_i} \boldsymbol{v}^{g_i}_{n+1/2} \cdot \boldsymbol{v}^{g_i}_{n+1/2} \tag{98}$$

$$v^{g_i}_{n+1/2} = \frac{\boldsymbol{x}^{g_i}_{n+1} - \boldsymbol{x}^{g_i}_{n}}{\Delta t} \tag{99}$$

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.

$$v^{g_i g_{i+1} ac}_{n+1/2} = \frac{A_{g_{i+1}ac}\boldsymbol{x}^{g_i}_{n+1} + A_{g_i ac}\boldsymbol{x}^{g_{i+1}}_{n+1} + A_{g_i c g_{i+1}}\boldsymbol{x}^{c}_{n+1} - A_{g_i a g_{i+1}}\boldsymbol{x}^{a}_{n}}{6\mathcal{V}_{g_i g_{i+1} ac}} \tag{100}$$

$$K^{g_i(III)}_{n,n+1} = m^n_{g_i g_{i+1} ac} \boldsymbol{v}^{g_i g_{i+1} ac}_{n+1/2} \cdot \boldsymbol{v}^{g_i g_{i+1} ac}_{n+1/2} \tag{101}$$

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

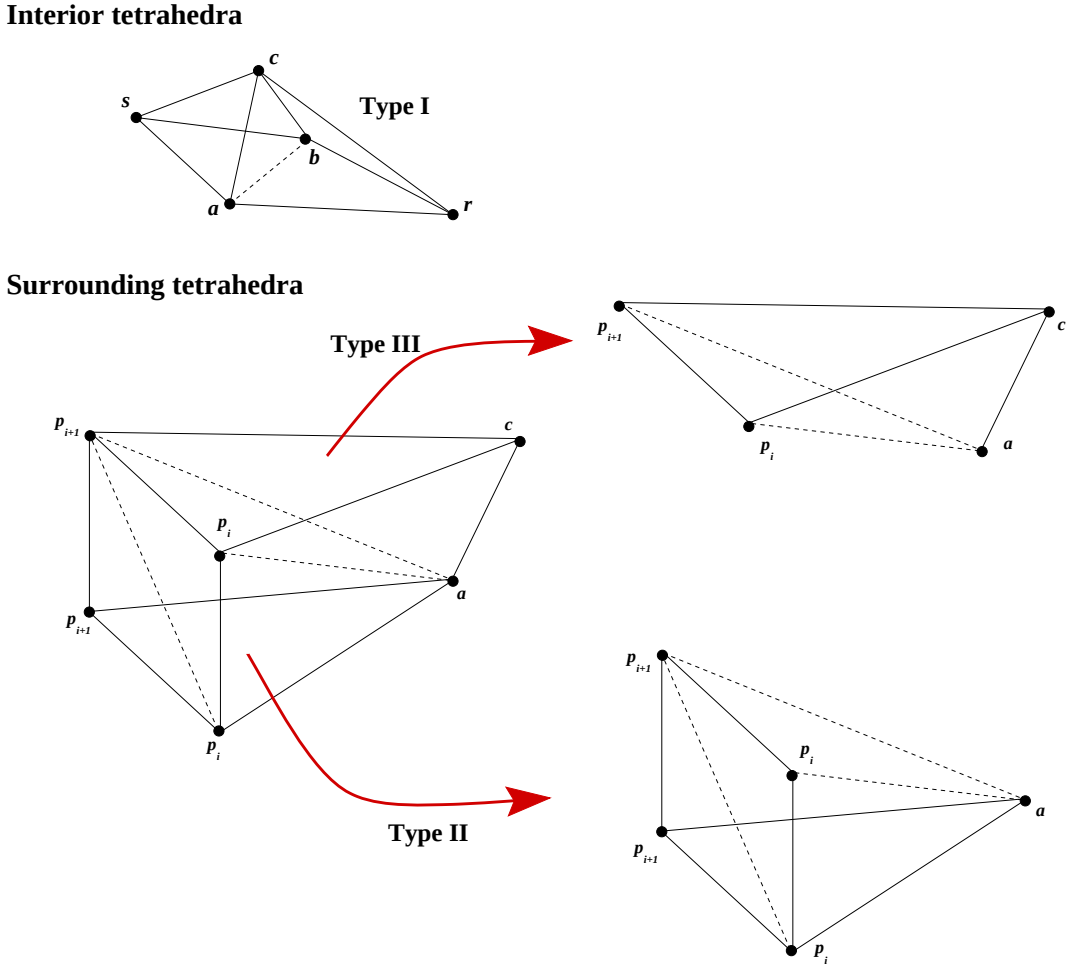

Figure 9. The subdivision of the space-time volume into different types of tetrahedra.

together, and the sum is expressed by the following approximation :

$$\sum_{g_i} K_{n,n+1}^{g_i(III)} = \frac{\Delta t}{2} m_c^{n+1} \boldsymbol{v}_{n+1/2}^* \cdot \boldsymbol{v}_{n+1/2}^* \tag{102}$$

$$\boldsymbol{v}_{n+1/2}^* = \frac{1}{\Delta t m_c^{n+1}} \left[m_c^* \boldsymbol{x}_{n+1}^c + \sum_i \Delta m_{g_i} \boldsymbol{x}_{n+1}^{g_i} - m_a^* \boldsymbol{x}_n^a - m_b^* \boldsymbol{x}_n^b \right] \tag{103}$$

$$m_c^* = m_c^{n+1} - \frac{m_{ab}^n}{3} \tag{104}$$

$$\Delta m_{g_i} = m_{g_i}^{n+1} - m_{g_i}^n \tag{105}$$

$$m_a^* = m_a^n - \frac{m_{ab}^n}{3} + M_{arcs} \tag{106}$$

$$m_b^* = m_b^n - \frac{m_{ab}^n}{3} + M_{bscr} \tag{107}$$

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 100. In addition, since the neighboring nodes are not moved, nor are any neighboring edge allowed to be collapsed, the mass m_c^{n+1} is known *apriori*. Thus the net Kinetic Energy integral becomes:

$$\begin{aligned} K_{n,n+1} &= K^I_{n,n+1} + K^{II}_{n,n+1} + K^{III}_{n,n+1} \\ &= \frac{\Delta t}{2}\frac{m^n_{ab}}{3}\boldsymbol{v}^c_{n+1/2}\cdot\boldsymbol{v}^c_{n+1/2} \quad + \quad \frac{\Delta t}{2}m_c^{n+1}\boldsymbol{v}^*_{n+1/2}\cdot\boldsymbol{v}^*_{n+1/2} \\ &+ \quad \sum_i \frac{\Delta t}{2}m^n_{g_i}\boldsymbol{v}^{g_i}_{n+1/2}\cdot\boldsymbol{v}^{g_i}_{n+1/2} \end{aligned} \tag{108}$$

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

$$(\forall\ g_i \neq r) \qquad m^n_{g_i}\ \boldsymbol{v}^{g_i}_{n+1/2} - m^n_{g_i}\ \boldsymbol{v}^{g_i}_{n-1/2} = \Delta t\ \left(\boldsymbol{F}^{g_i}_n - \boldsymbol{T}^{g_i}_n\right) \tag{109}$$

$$(1-\xi-\eta)\ \frac{m^n_{ab}}{3}\ \boldsymbol{v}^c_{n+1/2} + m^n_r\ \boldsymbol{v}^r_{n+1/2} - m^n_r\ \boldsymbol{v}^r_{n-1/2} = \Delta t\ \left(\boldsymbol{F}^r_n - \boldsymbol{T}^r_n\right) \tag{110}$$

$$\xi\ \frac{m^n_{ab}}{3}\ \boldsymbol{v}^c_{n+1/2} + m^*_a\ \boldsymbol{v}^*_{n+1/2} - m^n_a\ \boldsymbol{v}^a_{n-1/2} = \Delta t\ \left(\boldsymbol{F}^a_n - \boldsymbol{T}^a_n\right) \tag{111}$$

$$\eta\ \frac{m^n_{ab}}{3}\ \boldsymbol{v}^c_{n+1/2} + m^*_b\ \boldsymbol{v}^*_{n+1/2} - m^n_b\ \boldsymbol{v}^b_{n-1/2} = \Delta t\ \left(\boldsymbol{F}^b_n - \boldsymbol{T}^b_n\right) \tag{112}$$

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

$$(\forall\ j = g_i, a, b) \qquad \boldsymbol{R}^j_n = m^n_j\ \boldsymbol{v}^j_{n-1/2} + \Delta t\ \left(\boldsymbol{F}^j_n - \boldsymbol{T}^j_n\right) \tag{113}$$

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

$$(\forall\ g_i \neq r) \qquad m^n_{g_i}\ \boldsymbol{v}^{g_i}_{n+1/2} = \boldsymbol{R}^{g_i}_n \tag{114}$$

$$(1-\xi-\eta)\ \frac{m^n_{ab}}{3}\ \boldsymbol{v}^c_{n+1/2} + m^n_r\ \boldsymbol{v}^r_{n+1/2} = \boldsymbol{R}^r_n \tag{115}$$

$$\xi\ \frac{m^n_{ab}}{3}\ \boldsymbol{v}^c_{n+1/2} + m^*_a\ \boldsymbol{v}^*_{n+1/2} = \boldsymbol{R}^a_n \tag{116}$$

$$\eta\ \frac{m^n_{ab}}{3}\ \boldsymbol{v}^c_{n+1/2} + m^*_b\ \boldsymbol{v}^*_{n+1/2} = \boldsymbol{R}^b_n \tag{117}$$

Note here that Eqn. 114 is fully explicit, hence, $\boldsymbol{x}^{g_i}_{n+1}$ for all g_i except r are known. Now revisiting Eqn. 103 one can rewrite the expression for $\boldsymbol{v}^*_{n+1/2}$ using Eqn. 115 as:

$$\boldsymbol{v}^*_{n+1/2} = S_m(\xi,\eta)\boldsymbol{v}^c_{n+1/2} + \boldsymbol{W}(\xi,\eta) \tag{118}$$

$$\begin{aligned} \boldsymbol{W}(\xi,\eta) &= \frac{1}{\Delta t m_c^{n+1}}\left[\sum_i \Delta m_{g_i}\boldsymbol{z}^{g_i}_{n+1} - m^*_a\boldsymbol{x}^a_n - m^*_b\boldsymbol{x}^b_n + m^*_c\boldsymbol{x}^c_n + \Delta m_r\boldsymbol{x}^r_n\right] \\ &= \boldsymbol{W}_0 + \xi\ \boldsymbol{W}_\xi + \eta\ \boldsymbol{W}_\eta \end{aligned} \tag{119}$$

where,

$$\begin{aligned} \boldsymbol{z}_{n+1}^{g_i} &= \boldsymbol{x}_{n+1}^{g_i} \qquad (\forall\ g_i \neq r) \\ &= \frac{\Delta t\ \boldsymbol{R}_n^r}{m_n^r} \qquad (\ g_i = r) \\ S_m(\xi,\eta) &= \frac{m_{ab}^n}{3m_{n+1}^c m_n^r}\ \left[\frac{3m_c^* m_n^r}{m_{ab}^n} - (\Delta m_r(1-\xi-\eta))\right] \\ &= S_0 + S_\xi^1\xi + S_\eta^1\eta + S_\xi^2\xi^2 + S_\eta^2\eta^2 + S_{\xi\eta}^2\xi\eta \end{aligned} \tag{120}$$

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. 118 in Eqns. 116 & 117 the two equations are rewritten as:

$$K_a(\xi,\eta)\ \boldsymbol{v}_{n+1/2}^c\ +\ m_a^*(\xi,\eta)\ \boldsymbol{W}(\xi,\eta) \;=\; \boldsymbol{R}_n^a \tag{121}$$

$$K_b(\xi,\eta)\ \boldsymbol{v}_{n+1/2}^c\ +\ m_b^*(\xi,\eta)\ \boldsymbol{W}(\xi,\eta) \;=\; \boldsymbol{R}_n^b \tag{122}$$

where,

$$K_a(\xi,\eta) \;=\; \xi\ \frac{m_{ab}^n}{3} + m_a^*(\xi,\eta)\ S_m(\xi,\eta) \tag{123}$$

$$K_b(\xi,\eta) \;=\; \eta\ \frac{m_{ab}^n}{3} + m_b^*(\xi,\eta)\ S_m(\xi,\eta) \tag{124}$$

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

$$\boldsymbol{v}_{n+1/2}^c \;=\; \frac{\boldsymbol{R}_n^a - m_a^*(\xi,\eta)\ \boldsymbol{W}(\xi,\eta)}{K_a(\xi,\eta)} \tag{125}$$

$$\boldsymbol{f}(\xi,\eta) \;\equiv\; \frac{K_b}{K_a}\,(\boldsymbol{R}_n^a - \ m_a^*\ \boldsymbol{W}) + \left(\ m_b^*\ \boldsymbol{W} - \boldsymbol{R}_n^b\right) \;=\; 0 \tag{126}$$

Thus, a simple vector equation (126) 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 ($\boldsymbol{X}_{n+1}^c$) 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 (125,118 and 115). The position updates are obtained by the Eqns. 97 & 99. The momentum conserved in this time-step is of the form:

$$\boldsymbol{P}_{n,n+1} \;=\; \sum_j \boldsymbol{P}_{n,n+1}^j \tag{127}$$

$$\boldsymbol{P}_{n,n+1}^j \;=\; \begin{cases} m_n^j \boldsymbol{v}_{n+1/2}^j + \Delta m_j \boldsymbol{v}_{n+1/2}^*, & \text{for } j = g_i \\ \frac{1}{3} m_{ab}^n \boldsymbol{v}_{n+1/2}^j + (m_{n+1}^j - \frac{m_{ab}^n}{3})\boldsymbol{v}_{n+1/2}^*, & \text{for } j = c \end{cases} \tag{128}$$

$$\boldsymbol{H}_{n,n+1} \;=\; \sum_j \boldsymbol{x}_{n+1}^j \times \boldsymbol{P}_{n,n+1}^j \tag{129}$$

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:

$$m_{n+1}^{g_i} \boldsymbol{v}_{n+1/2}^{g_i} - m_n^{g_i} \boldsymbol{v}_{n+1/2}^{g_i} - \Delta m_{g_i} \boldsymbol{v}_{n+1/2}^* \;=\; \Delta t\ (\boldsymbol{F}_{n+1}^{g_i} - \boldsymbol{T}_{n+1}^{g_i}) \tag{130}$$

$$m_{n+1}^c \boldsymbol{v}_{n+1/2}^c - \frac{1}{3} m_{ab}^n \boldsymbol{v}_{n+1/2}^c - (m_{n+1}^c - \frac{m_{ab}^n}{3})\boldsymbol{v}_{n+1/2}^* \;=\; \Delta t\ (\boldsymbol{F}_{n+1}^c - \boldsymbol{T}_{n+1}^c) \tag{131}$$

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. There are various error estimates used in the literature based on solution gradients, functional outputs or from residual-error of various relations. 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 [68]. Elements with high values of this error, were chosen for adaptation.

Once the element was chosen, the 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, say ($\eta_i < 0.3$), were collapsed. Edges with edge-length-ratios closer to 1, say ($\eta_i > 0.7$), were swapped or split.

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.

Mesh adaptation was performed as a sequence of all the mesh operations (diagonal swapping, edge splitting, edge collapsing or node-movement) described above. All mesh operations 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). Diagonal swapping, node movement, edge-splitting and edge-collapsing were attempted in this sequence at every subsequent (or alternate) timestep pair. Depending on the need, 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.

The adaptation criteria used for the present mesh operations were chosen for their simplicity of implementation and were found to be quite effective for the cases discussed in this paper. Further development of the mesh adaptation criteria, is required for generic implementations. In the next section, performance of the final mesh-adaptive solver is demonstrated using examples from rapid dynamics of hyperelastic bodies.

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 10 (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. A simple mesh adaptation criteria based on the mesh edge skewness and elemental stress was used. Figure 10, demonstrates how the mesh was refined where the mesh skewness and stresses were relatively larger. The center of mass was initially at

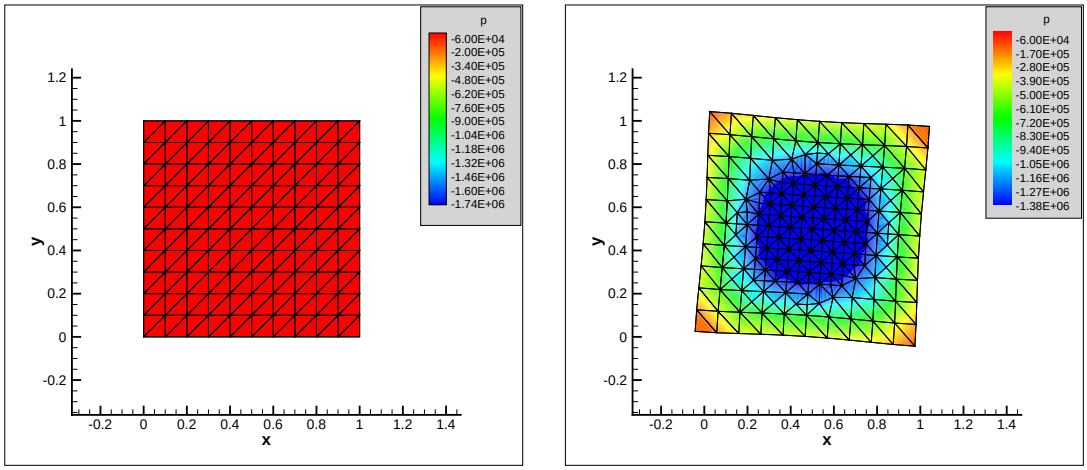

Figure 10. A Spinning plate simulation with adaptation.

$\boldsymbol{X} = 0.5, \boldsymbol{Y} = 0.5$, and as a consequence of conservation of linear momentum $\boldsymbol{P}_{n,n+1}$, the center of mass would be expected to remain at the same location. Although due to node movement, some minor fluctuation in the center of mass location could be expected. The angular momentum $\boldsymbol{H}_{n,n+1}$ would also be expected to remain constant during the time integration.

As expected, the momentum remains conserved exactly, throughout the simulation as shown in Figure 11. The momentum 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. There was no noticeable change in the center of mass, as shown in Figure 12, which was also expected. Figure 13 shows the energy history during the simulation. Although there are slight fluctuations in the energy behavior, initially, but with time, the energy remains more or less bounded, with no significant rise or dissipation as shown in Figure 13.

Figure 14, shows the adaptation history, with the number of operations conducted and the total number of nodes and elements during the course of adaptation.

5.2. An oscillating ring

A unit thickness circular ring, made up of nearly incompressible hyperelastic material (rubber) ($E = 1.7 \times 10^7\,\text{Pa}$, $\nu = 0.48$ and $\rho = 10^3\,\text{kg/m}^3$) 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 15 (left). The ring was stretched as shown in figure 15 (right), at time $t = 0$ and

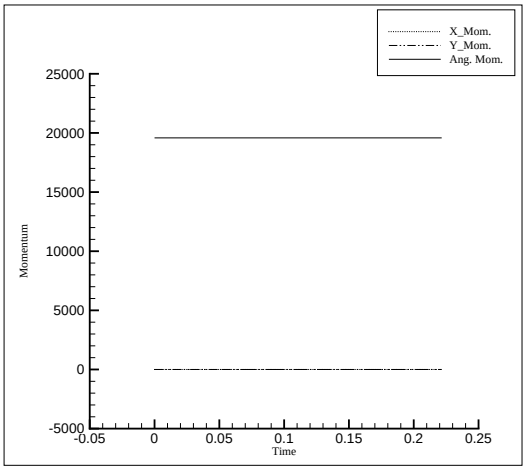

Figure 11. Linear and angular momentum history

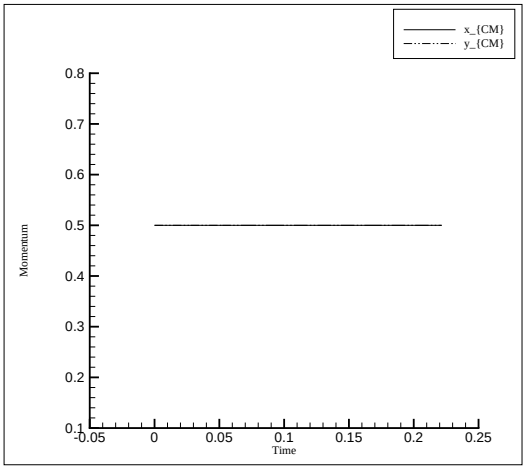

Figure 12. Location of center of mass.

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 16 - 20 show the material (reference) configurations and spatial (deformed) configurations at intermediate time steps. Figure 21 demonstrates the conservation of linear and angular momentum over time.

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.

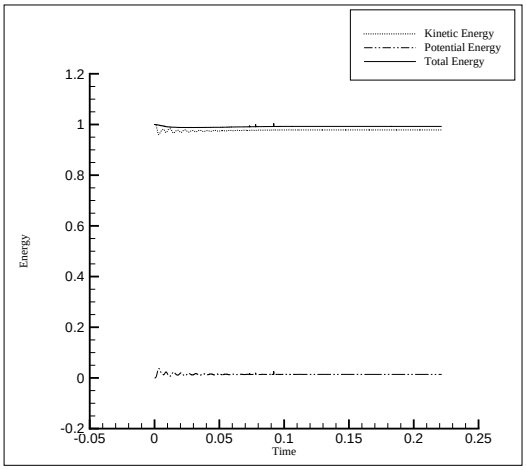

Figure 13. Energy history.

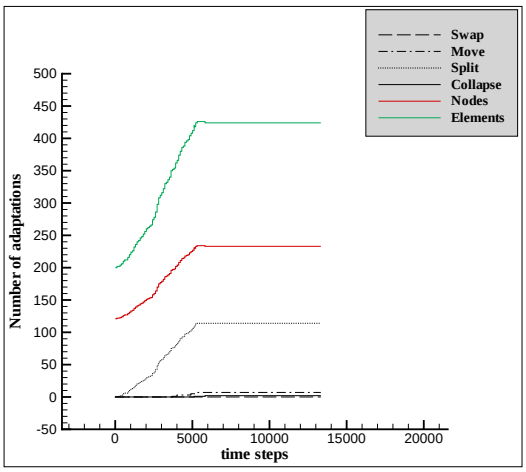

Figure 14. Mesh Adaptation history.

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.

$$\boldsymbol{P}'_{n,n+1} = \boldsymbol{P}_{n,n+1} - \sum_{0}^{n} \left[\left(\int_{V_0} \rho_0 \boldsymbol{g}_n dV_0 \right) + \sum_{a \in \Gamma} \boldsymbol{R}^a_{n+1} \right] \Delta t_n \tag{132}$$

$$\boldsymbol{H}'_{n,n+1} = \boldsymbol{H}_{n,n+1} - \sum_{0}^{n} \left[\left(\int_{V_0} \rho_0 \boldsymbol{x}_n \times \boldsymbol{g}_n dV_0 \right) + \sum_{a \in \Gamma} \boldsymbol{x}_n \times \boldsymbol{R}^a_{n+1} \right] \Delta t_n \tag{133}$$

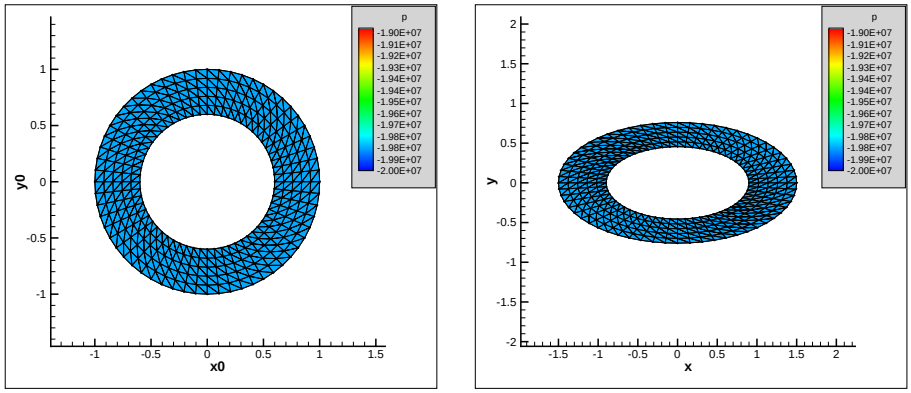

Figure 15. Ring at time $t = 0\,s$, with material(left) and spatial(right) configurations.

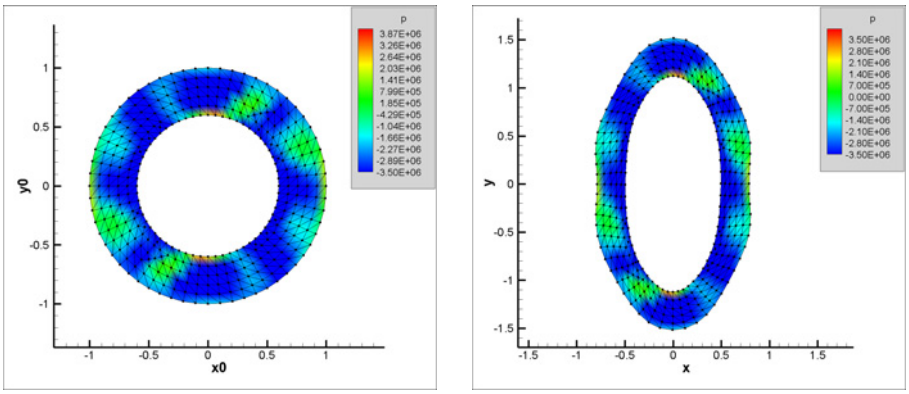

Figure 16. Ring at time $t = 0.05\,s$, with material(left) and spatial(right) configurations.

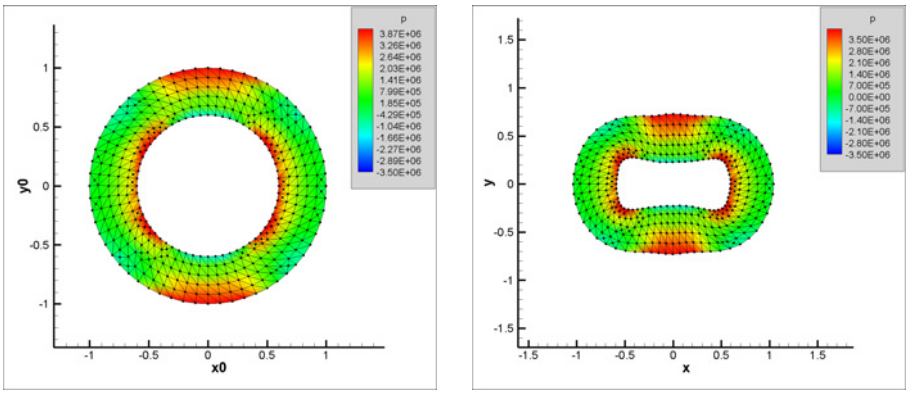

Figure 17. Ring at time $t = 0.10\,s$, with material(left) and spatial(right) configurations.

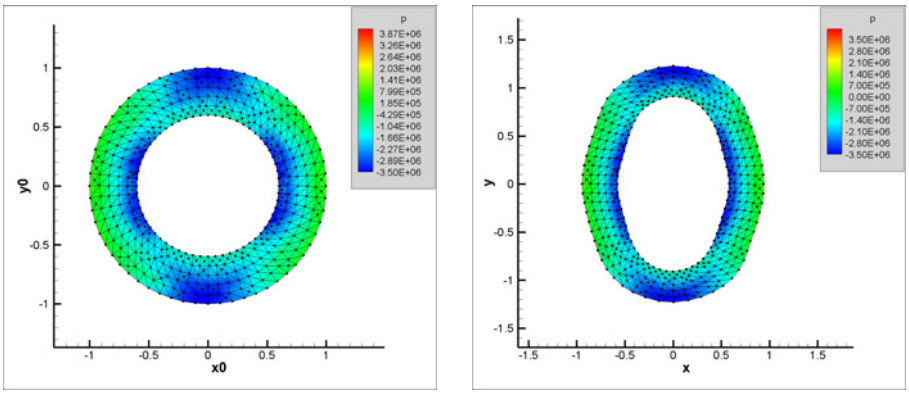

Figure 18. Ring at time $t = 0.15\,s$, with material(left) and spatial(right) configurations.

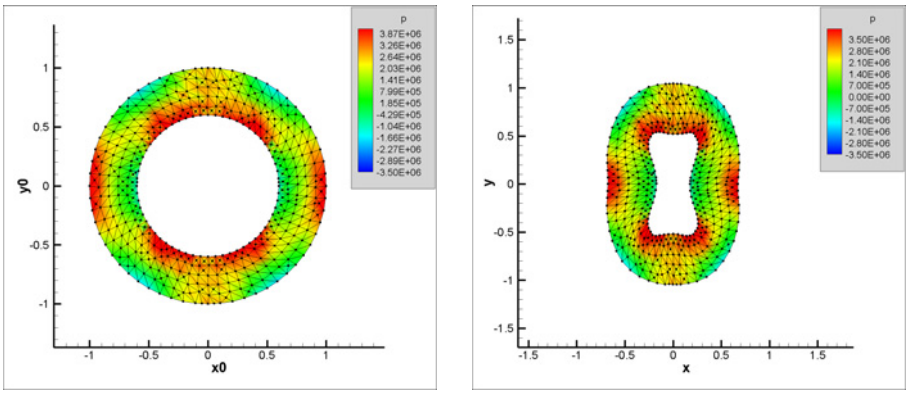

Figure 19. Ring at time $t = 0.17\,s$, with material(left) and spatial(right) configurations.

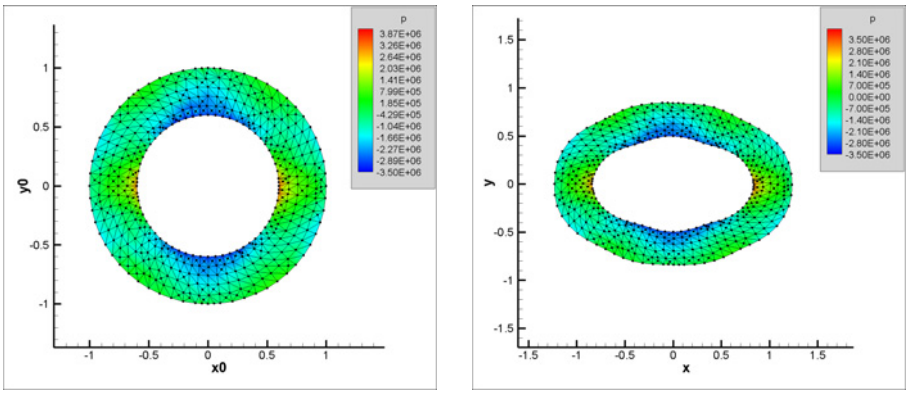

Figure 20. Ring at time $t = 0.20\,s$, with material(left) and spatial(right) configurations.

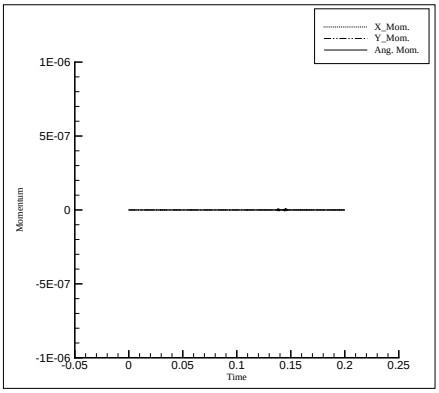

Figure 21. The momentum history for the oscillating ring.

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.

In the Figure 22 a square steel plate, with material properties ($E = 2.1 \times 10^{10}\,\mathrm{Pa}$, $\nu = 0.3$ and $\rho = 7 \times 10^3\,\mathrm{kg/m}^3$) is pulled rapidly by $\boldsymbol{v}_{\mathrm{pull}} = 40m/s$ at its top surface, and reaches thrice its length within 0.05seconds.

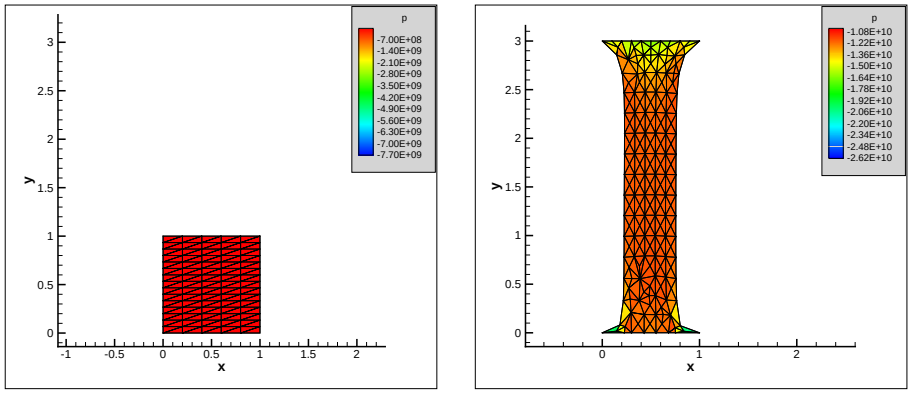

Figure 22. A Tensile test specimen (left) pulled to thrice its length (right).

Mesh adaptation was employed in the simulation, and the net momentum was conserved as is shown in Figure 23.

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

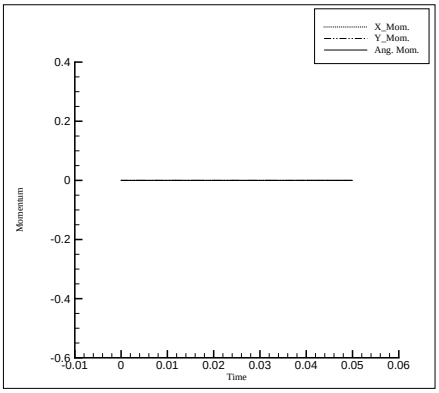

Figure 23. The Modified Momentum history for the Tensile test

the top half with a prescribed punch velocity ($v_{\text{punch}} = 2$ m/s) as shown in Figure 24. Here

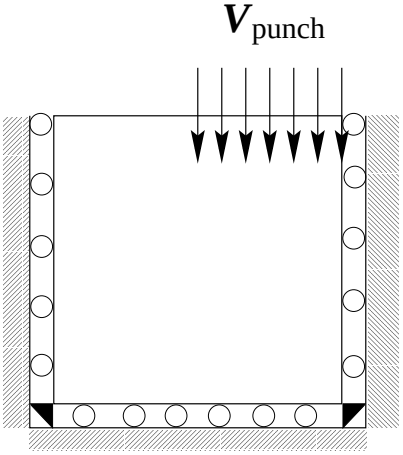

Figure 24. A schematic figure of the punch test case, showing boundary conditions.

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 25.

The Modified momentum remains conserved as shown in the figure 25.

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

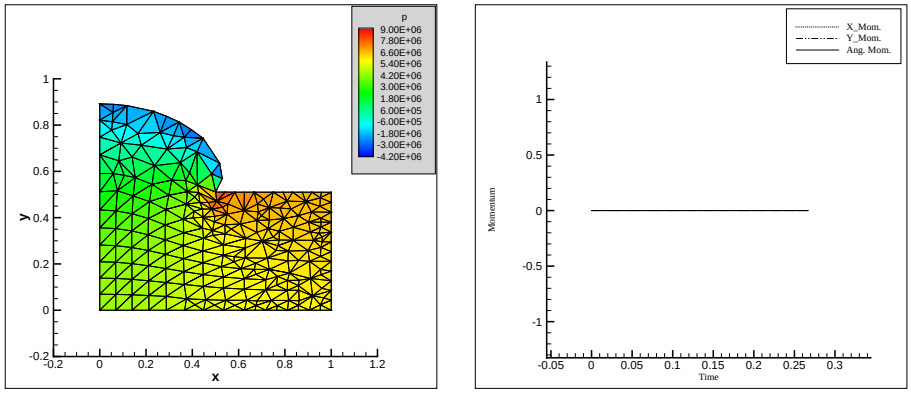

Figure 25. Mesh adaptation for a punch problem and the momentum history.

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 26 The material properties of the plate

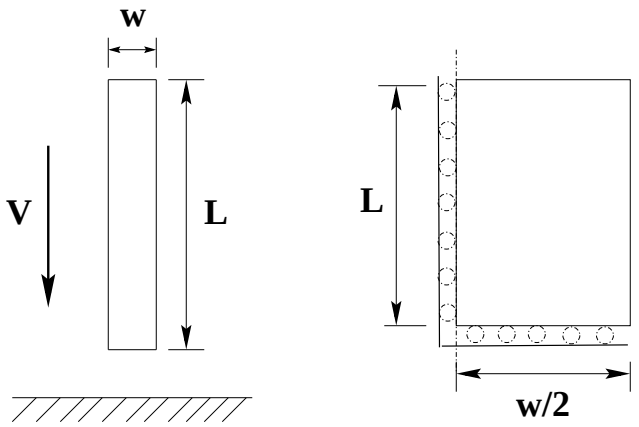

Figure 26. Schematic diagram of the plate impact problem.

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 27 to 30 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 30 shows adaptation near the contact where the mesh undergoes the most skewness. The plate touches the wall at 14 μs. In figure 27 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 inside, where the mesh is adapted. All the kinetic energy of the plate is almost converted to internal potential energy by 90 μs, almost sticking to the wall, as shown in figure 28. At 120 μs the plate begins to

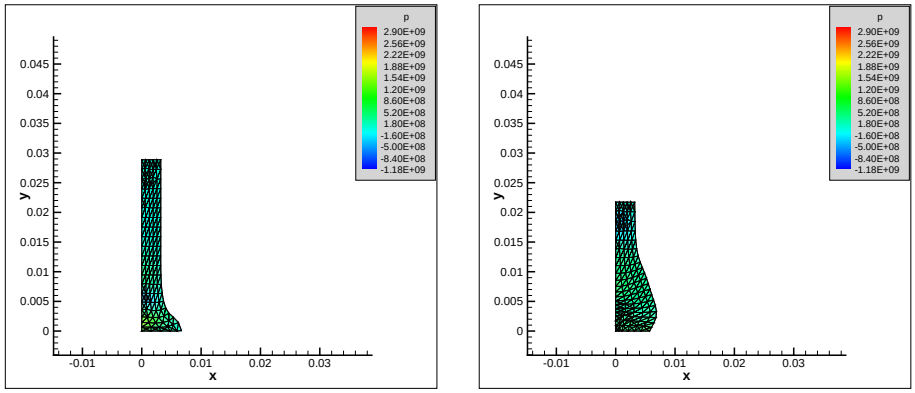

Figure 27. The plate at $t = 30\,\mu$s (left) and $t = 60\,\mu$s (right).

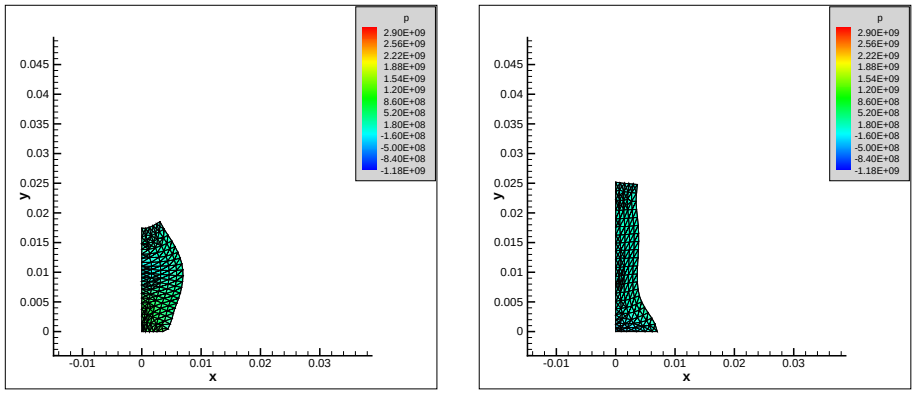

Figure 28. The plate at $t = 90\,\mu$s (left) and $t = 120\,\mu$s (right).

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 29. At roughly $180\,\mu$s, the plate leaves the rigid wall (bounce-off motion).

The modified momentum in this case remained constant, as shown in figure 31.

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

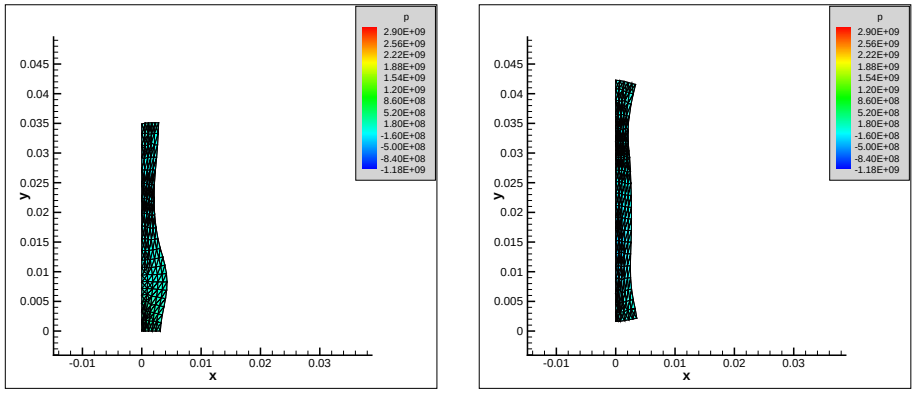

Figure 29. The plate at $t = 150\,\mu$s (left) and $t = 180\,\mu$s (right).

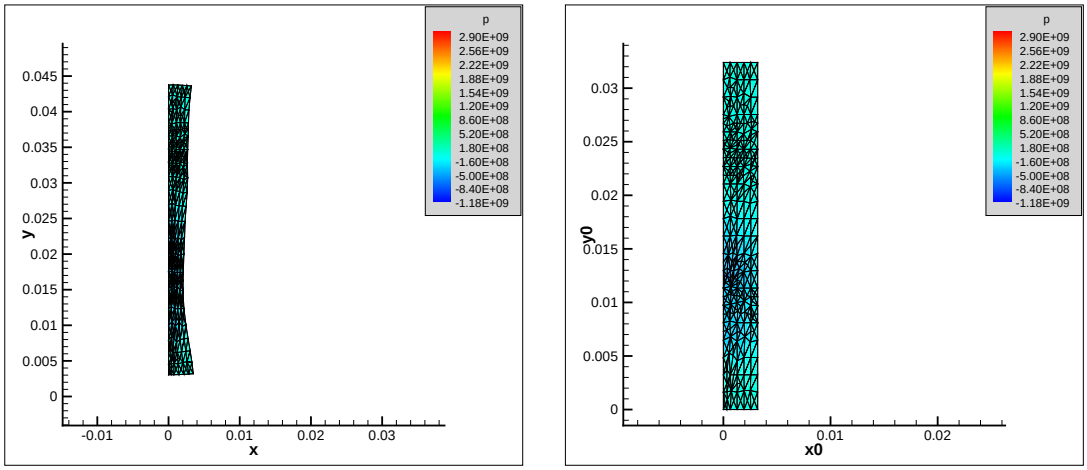

Figure 30. Deformed configuration of the plate at $t = 194\,\mu$s (left) and the corresponding mesh in the reference configuration (right).

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.

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.

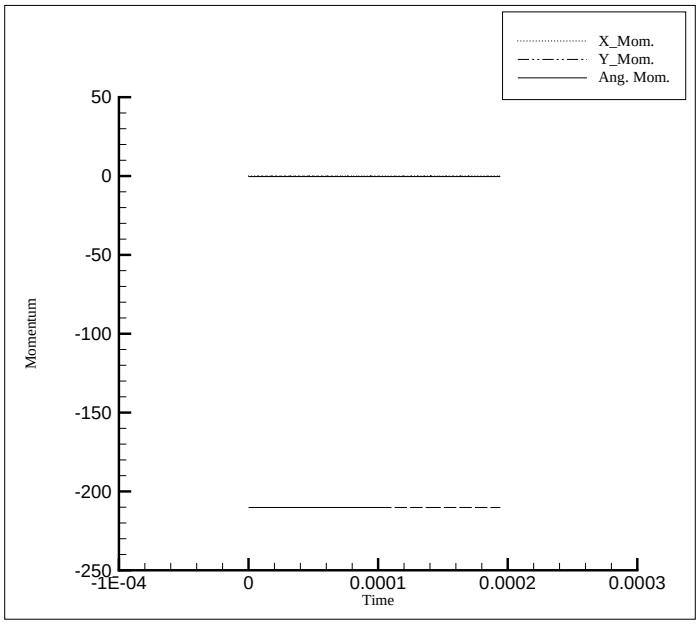

Figure 31. Modified Momentum history of the plate

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. I. Romero and L. M. Lacoma. A methodology for the formulation of error estimators for time integration in linear solid and structural dynamics. *Int. J. Numer. Meth. Engng.*, 66:635–660, 2006.
52. Weizhang Huang. Variational mesh adaptation: isotropy and equidistribution. *J. Comput. Phys.*, 174(2):903–924, 2001.
53. Weizhang Huang and Weiwei Sun. Variational mesh adaptation ii: error estimates and monitor functions. *J. Comput. Phys.*, 184(2):619–648, 2003.
54. 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.
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. R. Lohner, K. Morgan, and O. C. Zienkiewicz. An adaptive finite element procedure for compressible high speed flows. *Comput. Methods Appl. Mech. Engrg.*, 51:441–465, 1985.
58. R. Lohner. Adaptive remeshing for transient problems. *Comput. Methods Appl. Mech. Engrg.*, 75:195–214, 1989.
59. J. Peraire, M. Vahdati, K. Morgan, and O. C. Zienkiewicz. Adaptive remeshing for compressible flow computations. *J. Comp. Phys.*, 72:449–466, 1987.
60. O. Hassan, E. J. Probert, K. Morgan, and J. Peraire. Mesh generation and adaptivity for the solution of compressible viscous high speed flows. *Int. Journal. for Numerical Methods in Engrg.*, 38:1123–1148, 1995.
61. E. J. Probert, O. Hassan, and K. Morgan. An adaptive finite element method for transient compressible flows with moving boundaries. *Int. Journal. for Numerical Methods in Engrg.*, 32:751–765, 1991.
62. J. Stewart, A. R. Thareja, R. R. Wieting, and K. Morgan. Application of finite elements and remeshing techniques to shock interference on a cylindrical leading edge. *AIAA Paper*, 87-0032, 1988.
63. O. C. Zienkiewicz and J. Wu. Automatic directional refinement in adaptive analysis of compressible flows. *Int. Journal. for Numerical Methods in Engrg.*, 37:2189–2210, 1994.
64. M. Parashivoiu, J. Peraire, and A. T. Patera. A posteriori finite element bounds for linear functional outputs of elliptic partial differential equations. *Comput. Methods Appl. Mech. Engrg.*, 150:289–312, 1997.
65. D.A. Venditti and D.L. Darmofal. Anisotropic grid adaptation for functional outputs: application to two-dimensional viscous flows. *J. of Comp. Phys.*, 187:22–46, 2003.
66. Javier Bonet and Richard D. Wood. *Nonlinear continuum mechanics for finite element analysis*. Cambridge University Press, 1997.
67. Gerhard A. Holzapfel. *Nonlinear Solid Mechanics*. John Wiley & Sons Ltd., 2000.
68. 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.
69. Ted Belytschko, Wing Kam Liu, and Brian Moran. *Nonlinear Finite Elements for Continua and Structures*. John Wiley & Sons Ltd., 2000.