A Framework for Fast Computation of Hyper-Elastic Materials Deformations in Real-Time Simulation of Surgery

François Goulette, Safwan Chendeb

Robotics Lab, Mines Paris, 60 bd Saint Michel, 75272 Paris Cedex 06, France
{goulette, chendeb}@ensmp.fr

Abstract. Real-time simulation of surgery requires fast and realistic modeling of deformable objects. However today, few models are available, they are still time costly and limited in number of tetrahedra by algorithm complexity. We present here a framework to design algorithms adapted to any kind of hyperelastic models, which include Saint Venant-Kirchhoff, Neo-Hookean and Mooney-Rivlin, handling large deformations and large displacements. Moreover, the algorithm complexity is linear in the number of tetrahedra; a comparison in number of operations performed with an existing algorithm on a similar case shows a gain of more than 40 times faster.

1 Introduction

Fast and realistic modeling of deformable objects is an issue for the development of medical surgery simulators. Good numerical models should handle in real-time deformations coming from interactions with the user and its virtual environment, constraints, large deformations, large displacements, realism with several material constitutive laws, easy cutting and topological changes. Real-time refers to a computation time of the discrete differential equations small enough to allow a reasonable refreshment rate, e.g. 30 Hz for visual display. Several computational methods and models have been developed to simulate real-time soft tissue deformations. Many of these models have addressed the issue of computing the internal elastic force of the material, related to its intrinsic material properties. Some researchers have been interested in mass-spring models [1], [2], due to their simplicity of implementation and their low computation complexity; they also handle naturally large deformations, large displacements, and cutting. On the other hand they suffer from a lack of realism, which lead to further research on extensions of the model, or correct tuning of the parameters for better biomechanical realism: [3], [4], [5]. Other people have preferred to derive discrete computational algorithms from the equations of Continuum Mechanics, in order to obtain real-time computations: based on the Boundary Element Method (BEM) [6], the Finite Difference Method (FDM) [7] and the Finite Element Method (FEM) [8], [9]. However, the heavy complexity of these methods makes computation time a real challenge; at first, methods have been developed with the simplifying assumptions of small deformations and displacements (linear elasticity

framework); this has lead e.g. to the mass-tensor model [10]; this model has been developed further to handle large deformations and large displacements with a "Saint Venant-Kirchhoff" constitutive law [11], [12], with more realism but also more complexity and operations to perform. As a summary, one can say that, if progresses have been made, existing models still handle few constitutive laws (linear elasticity - limited to small deformations-, and Saint Venant-Kirchhoff for large deformations), and suffer from a polynomial complexity that makes impossible to compute large meshes.

For these reasons, in this paper we propose a framework to design fast algorithms to compute the elastic force field for any hyper-elastic model, handling large deformations, large displacements, easy cutting and topological changes in the mesh. The algorithms are designed under the P1-finite element approximation in homogeneous isotropic cases. Hyper-elastic models include the Saint Venant-Kirchhoff constitutive law (used in mass-tensor), and other important hyper-elastic constitutive laws such as Neo-Hookean, Mooney-Rivlin, Fung and Demiray, or Ogden, that are not supported today by real-time algorithms. We have chosen to call this approach "Hyper-Elastic Mass-Link" (HEML), for the following reasons. "Link", because forces at a given node are given as a sum of forces proportional to the links (vectors) to all connected neighbours; the formulation of the modulus of these incident forces depend only from the square lengths of the links of adjacent tetrahedra. "Mass": as in mass-spring or mass-tensor, masses are affected to the mesh nodes, used in the discrete differential equations. "Hyper-Elastic", because the framework presented may be used to design algorithms for computation of any hyper-elastic material. As a matter of fact, it also includes the mass-spring model as a special, degenerate (non hyper-elastic) case. We detail in Section 2 the methodology to obtain Hyper-Elastic Mass-Link models, in general and for several hyper-elastic constitutive laws. In Section 3, we present an implementation made in the Saint Venant-Kirchhoff case, and a comparison with the mass-tensor algorithm which shows an improved performance in number of operations.

2. "Hyper-Elastic Mass-Links" Models

2.1 Notations

We consider a discrete approximation of the material geometry based on a tetrahedral mesh, and reasoning is made for each tetrahedron (Figure 1). Without loss of generality, a numbering is chosen for the four vertices of a given tetrahedron, which are denoted X_i and x_i ($0 \le i \le 3$) in initial and deformed state. The six edge vectors are denoted VE_i (respectively ve_i) ($1 \le i \le 6$). Any three edge vectors out of six are enough to express the others, we denote $\boldsymbol{ME}$ (resp. $\boldsymbol{me}$) the matrix: $\boldsymbol{ME} = (VE_1, VE_2, VE_3)$ (matrices are denoted in bold typeface). Considering non-degenerate tetrahedra in initial state, $\boldsymbol{ME}$ is invertible. We denote l_i the lengths of edges ve_i (resp. L_i for VE_i), l (resp. L) the vector of the six square lengths, and $\Delta l = l - L$ the vector of differences. The volume of tetrahedron in deformed (resp. initial) state is denoted v (resp. V). The deformation function from initial to current state is Φ, and $\boldsymbol{F}$ the gradient of deforma-

tion: $\boldsymbol{F}(X,t) = \nabla\Phi = \partial x/\partial X$. The right Cauchy-Green deformation tensor $\boldsymbol{C}$ and the Green-Lagrange strain tensor $\boldsymbol{E}$ are: $\boldsymbol{C}(X,t) = \boldsymbol{F}^T\boldsymbol{F}$, $\boldsymbol{E} = (\boldsymbol{C} - \boldsymbol{I})/2$. The "three invariants" of tensor $\boldsymbol{C}$ are denoted C_I, C_{II}, C_{III}.

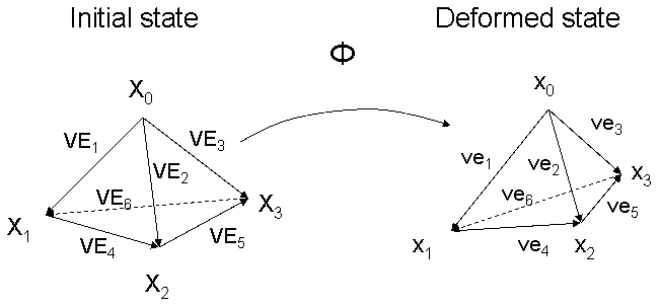

Fig. 1. Tetrahedron in initial and deformed state

2.2 Hyper-Elasticity in P1 Approximation: Energy and Forces

Hyperelasticity [13] means that there exists a volumetric energy function W from which derives the stress tensor, function of position x and deformation gradient $\boldsymbol{F}$. We consider a homogeneous, isotropic, hyper-elastic material. As a homogeneous material, the energy function W depends only from the deformation gradient $\boldsymbol{F}$. Additionally, the two principles of "material indifference", imposed by coherence with physics laws, and isotropy, lead to the fact that W depends only from the tensor $\boldsymbol{C}$; moreover it depends only from the three invariants of $\boldsymbol{C}$: $W(x, \boldsymbol{F}) = W(\boldsymbol{C}) = W(C_I, C_{II}, C_{III})$.

In the "P1 approximation" of finite elements, mesh elements are tetrahedral; the approximation states that the deformation gradient tensor $\boldsymbol{F}(x)$ is constant over a given tetrahedron T_k; hence also for tensor $\boldsymbol{C}$. We can decompose the energy W over each tetrahedron T_k, the total energy of the material being the sum over all tetrahedra. It can be demonstrated (see Appendix) that, under the P1 approximation, the value of $\boldsymbol{C}$ depends linearly from the vector l (edge square lengths) and from initial state. Hence the energy W_k of tetrahedron T_k depends only from l_k and initial state, or alternatively it depends only from vertices positions and initial state:

$$W\_k = W\_k(l\_k) = W\_k(x_0, x_1, x_2, x_3). \quad \textbf{(1)}$$

As forces derive from energy, at each node of the mesh the internal elastic force is the sum of forces derived from the energy of all incident tetrahedra. For each tetrahedron T_k, the elastic force at a vertex X_j is the derivative of energy W_k over the vertex position: $\forall j, 0 \le j \le 3, \partial W_k/\partial X_j = (\partial W_k/\partial l) \times (\partial l/\partial X_j)$. It can be demonstrated (see Appendix) that the derivative of l over the vertices positions is a linear expression of the matrix $\boldsymbol{me}$ of edge vectors, with constant 6x3 matrices $\boldsymbol{DLM}$, leading to the following linear formulation of forces for each vertex X_j of tetrahedron T_k:

$$\forall j, 0 \le j \le 3, F^T_{j\_k} = \frac{\partial W\_k}{\partial X_j} = \frac{\partial W\_k}{\partial l}(l\_k) \times \boldsymbol{DLM_j} \times me\_k^T \quad \textbf{(2)}$$

2.3 Formulation for various hyperelastic materials

We have used this approach (Equations 1, 2) to derive elastic forces for various specific materials: Saint Venant-Kirchhoff, Neo-Hookean, Mooney-Rivlin, Fung and Demiray. Saint Venant-Kirchhoff is an hyperelastic material, which is a natural extension to large deformations of linear elastic constitutive law. The volumetric energy function is usually formulated with the Green-Lagrange tensor $\boldsymbol{E}$, and the Lamé coefficients λ and μ. Neo-Hookean [15] and Mooney-Rivlin [16] are popular non-linear hyperelastic materials; Fung and Demiray [17], [18] is a model that has already been used for biomechanical application [19]. For each kind of hyper-elastic material, we use for constitutive law the formulation of its volumetric density of energy W, expressed with respect to the tensor $\boldsymbol{C}$, its three invariants C_I, C_{II}, C_{III}, or alternatively the tensor $\boldsymbol{E}$, and we express it with respect to the edges square lengths l (Equation 1), or alternatively Δl. Then we derive a formulation of forces with respect to l (Equation 2) or Δl, and $\boldsymbol{me}$ (Appendix). We present in Table 1 the energy density and forces derived at the vertices of a given tetrahedron, for each of these materials.

Table 1. Energy and forces for various hyper-elastic materials – for one tetrahedron

Material	Energy density W	Force at vertex j
General type	$f(C_I, C_{II}, C_{III}) = f(l)$	$\sum_{i\in\{I,II,III\}} \frac{\partial f}{\partial C_i}(l) \times \frac{\partial C_i}{\partial X_j}(l, \boldsymbol{me})$
Saint Venant-Kirchhoff	$\frac{\lambda}{2}(Tr(\boldsymbol{E}))^2 + \mu Tr(\boldsymbol{E}^2) = \Delta l^T \underbrace{\left(\frac{\lambda}{8} VTr^T VTr + \frac{\mu}{4}\boldsymbol{MTr}\right)}_{\boldsymbol{MW_{STV-K}}} \Delta l$	$\Delta l^T (\boldsymbol{MW_{STV-K}} \times \boldsymbol{DLM_j}) \times \boldsymbol{me}^T$
Neo-Hookean (incompressible)	$K_1 \times C_I = K_1 (VTr \times l)$	$K_1 (VTr \times \boldsymbol{DLM_j}) \times \boldsymbol{me}^T$
Mooney-Rivlin (incompressible)	$K_1 \times C_I + K_2 \times C_{II} = K_1 (VTr \times l) + K_2 l^T \underbrace{\left[\frac{1}{2}\left(VTr^T VTr - \boldsymbol{MTr}\right)\right]}_{\boldsymbol{MW_{M-R}}} l$	$K_1 (VTr \times \boldsymbol{DLM_j}) \times \boldsymbol{me}^T + K_2 l^T (\boldsymbol{MW_{M-R}} \times \boldsymbol{DLM_j}) \times \boldsymbol{me}^T$
Fung and Demiray (incompressible)	$\frac{\mu}{2\gamma}\exp(\gamma(C_I - 3) - 1) = \frac{\mu}{2\gamma}\exp(\gamma VTr \times \Delta l - 1)$	$\frac{\mu}{2}\exp(\gamma VTr \times \Delta l - 1) \times (VTr \times \boldsymbol{DLM_i}) \times \boldsymbol{me}^T$

Many bodies in biomechanical applications are mostly composed with water and should hence be considered as incompressible. In Saint Venant-Kirchhoff case, the material is compressible: incompressibility is a special, limit case, when the Poisson coefficient tends towards 0.5, which is in practice not possible for Lamé coefficients that become infinite. The Saint Venant-Kirchhoff model has also a serious drawback to be mentioned, which is that it has a wrong limit in compressibility: the force tends

to zero when the material flattens, which is non physical at all! In the case of Neo-Hookean, Mooney-Rivlin and "Fung and Demiray", the constitutive laws indicated are given under the strict constraint of incompressibility. However there is no easy way to handle such constraint in the numerical solving. On a practical point of view, a possibility for all cases is to add to the energy function a term depending on the third invariant of C, which itself depends upon volume variation – which actually means a modification of the constitutive law.

3. Implementation for Saint Venant-Kirchhoff materials

3.1 Algorithm

In order to compute an elastic force field on a material from the equations presented above, one has to perform two kinds of operations: an initialization regarding the initial state of the material (geometry and material parameters); a computation of forces in deformed state, at each step of the numerical integration process. Concerning initialization, one has to compute, for each tetrahedron, four characteristic matrices from its initial state and the two Lamé coefficients λ, μ. Concerning the force field in deformed state, for each tetrahedron one has to compute the six edge vectors and square lengths, and then, using the four characteristic matrices, the four forces at each vertex using the corresponding equation given in Table 1. In the end one has to sum up, for each node of the mesh, the forces coming from incident tetrahedra.

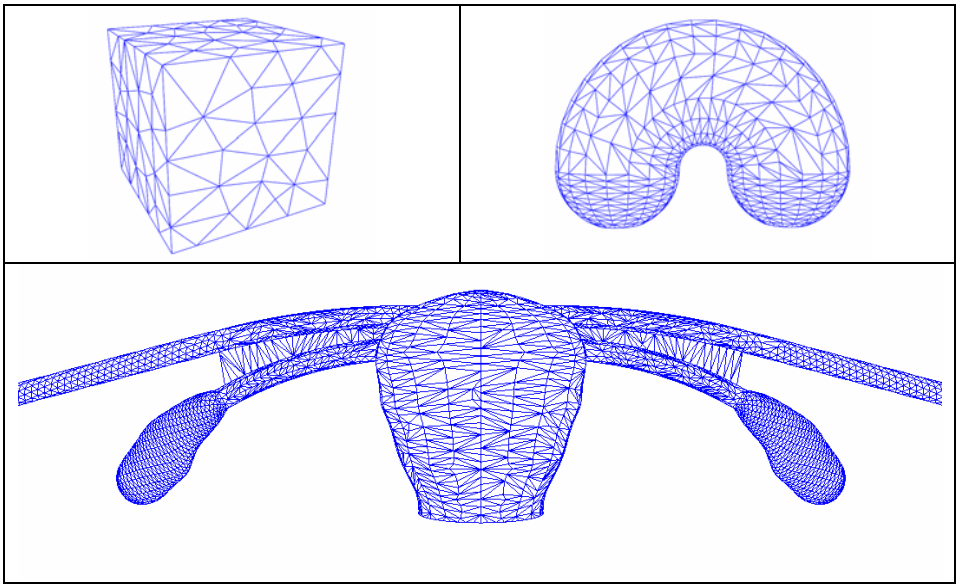

Fig. 2. Meshes used for experiments (irregular cube, artificial kidney, uterus)

We have implemented this algorithm on a PC computer, with the Visual C++ developing environment under Windows XP. The PC was a Pentium 4 at 3.2 Ghz, 1 Go

RAM. For a complete computation of material deformations, we add gravity, viscous forces (related to the velocities of nodes), and perform numerical integration with semi-implicit first-order Euler scheme. We present results based on non-optimized code, obtained on 5 different meshes: three cubes, two with regular and one with irregular tetrahedra, an artificial shape representing a kidney and a uterus model digitized from a plastic uterus used for the training of midwives (Figure 2). Mesh characteristics and average computation times per time step for the elastic force field are given in Table 2. It is noticeable that, for the uterus model, containing as many as 11 445 tetrahedra, the computation is feasible and can reach an average time of 20 ms.

Table 2. Models characteristics and computation times

	Reg. Cube1	**Reg. Cube2**	**Irreg.Cube**	**Kidney**	**Uterus**
Nodes	125	1000	151	829	4362
Tetrahedra	320	3645	407	2053	11445
Comp. time (ms)	0.25	3.5	0.34	2.1	20.0

3.2 Comparison with the "Mass-Tensors" algorithm

The Hyper-Elastic Mass-Link (HEML) algorithm for Saint Venant-Kirchhoff derives from the same formula as the Mass-Tensors algorithm adapted to large deformations. However the formulation is not the same; a noticeable difference is that HEML uses relative displacements of nodes within the mesh, while Mass-Tensor uses displacements relative to original positions, $U_j = x_j - X_j$. This may lead to numerical drawbacks with large displacements, when the numerical values of displacements are orders of magnitude (x100, x1000…) higher than positions. In order to reduce the number of operations in the expression of the Mass-Tensor algorithm, an arrangement is made, introducing matrices B_{pj}, vectors C_{jkl} and scalars D_{jklp}, and summing up at each node from all contributors, that is the node itself p and its connected neighbours N(p), leading to the following expression [12]:

$$F^p = \sum_{j \in \{p, N(p)\}} \boldsymbol{B_{pj}} U_j + 4 \sum_{j,k,l \in \{p, N(p)\}^3} D_{jklp} U_l U_k^T U_j + \sum_{j,k \in \{p, N(p)\}^2} 2\left(U_k \times U_j^T\right) C_{jkp} + \left(U_j^T \times U_k\right) C_{pjk} \qquad \textbf{(3)}$$

The complexity of Mass-Tensors is related to the number of connected edges, up to a power three as can be seen on the equation above; for HEML, the complexity is linear in the number of tetrahedra. Analyzing further the equations, we have determined the exact number of operations -multiplications and additions- needed by both algorithms, to compute the same force field over a mesh from the node positions. To perform a comparison we have determined the number of operations in the case of the five meshes presented above, and the results are presented on Table 3. We observe an impressive improvement with HEML, going for small meshes from a gain of more than 40 times, up to more than 100 and 200 with bigger meshes (uterus and kidney). It is also important to notice that the number of operations for other kinds of hyperelastic materials, like Neo-Hookean and Mooney-Rivlin, is of the same order of mag-

nitude as it is for Saint Venant-Kirchhoff, as one can see from the equations presented in Table 1.

Table 3. Number of operations (in thousands), and gain ratio, for Mass-Tensor and HEML

	Reg. Cube1	**Reg. Cube2**	**Irreg.Cube**	**Kidney**	**Uterus**
Mass-Tensor	2 999	36 198	3 935	96 292	318 707
HEML	70	787	89	449	2 500
Gain ratio	**43,0**	**46,0**	**44,3**	**214,3**	**127,5**

4 Conclusion

We have presented the "Hyper-Elastic Mass-Link" framework, a methodology for fast computation of deformable bodies that handles any kind of hyper-elastic material. Equations have been presented for four different materials (Saint Venant-Kirchhoff, Neo-Hookean and Mooney-Rivlin, Fung and Demiray) and the method has been explained to be extended to additional materials (e.g. Ogden materials [20]). It could also be extended to non homogeneity, and to anisotropy for e.g. transversally isotropic materials [12]. The complexity of the algorithms is linear in the number of tetrahedra. An implementation has been made with the Saint Venant-Kirchhoff constitutive law, handling the computation of a mesh of more than 10 000 tetrahedra in an average operating time of 20 ms, with non-optimized code. In that case, the algorithm presents several improvements over the mass-tensor algorithm: computations are made with respect to relative positions of nodes, and not to initial positions, which is numerically better for large displacements; the number of operations is much smaller, simulations on several meshes show improvements of more than 40 times.

Further work could be done on optimisation of implementation, to improve computation times, with e.g. specific coding for fast computation, or parallelisation; it would also be helpful to handle incompressibility issue and invertible elements, for example with signed penalty volumes or other adaptations of constitutive law [21]. Other improvements could be to evaluate the quality of P1 approximation -possibly using adaptive mesh refinements-, to model more precisely viscous terms, and to add plastic deformations.

Acknowledgments

This work was partly supported by the RNTS SIMV@L project of the French Minsitry of Research. The authors would also like to thank the CEMEF laboratory of Mines Paris for providing the kidney and uterus mesh models.

References

1. U. G. Kuhnapfel et al. : Endoscopic surgery training using virtual reality and deformable tissue simulation. Computers & Graphics, 24:671--682, 2000.
2. P. Meseure and C. Chaillou: A Deformable Body Model for Surgical Simulation. Journal of Visualization and Computer Animation, 11(4), Sept. 2000, pp. 197-208.
3. Van Gelder: An Approximate Simulation of Elastic Membranes by Triangulated Spring Meshes. In: Journal of Graphics Tools. v. 3, n. 2, pp. 21-42. (1998).
4. J. Jansson, J.S.M. Vergeest: A discrete mechanics model for deformable bodies. Journal of Computer-Aided Design, Vol. 34, Nr. 12, 1-1-2002.
5. A. Maciel et al. : Deformable Tissue Parameterized by Properties of Real Biological Tissue. In Proc. of the Int. Symposium on Surgery Simulation and Soft-Tissue Modeling, 2003.
6. C. Monserrat, V. Hermandez, M. Alcaniz, M.C. Juan, V. Grau. : A new approach for real time simulation of tissue deformations in surgery simulation. Computer Methods and Program in Biomedicine, pp. 75-84, 2001.
7. Debunne, G., Desbrun, M., Cani, M.P., and Barr, A.H., Dynamic Real-Time Deformations using Space & Time Adaptive Sampling. Siggraph.01, Computer Graphics annual conference series, Los Angeles, August 2001.
8. J. Berkley, S. Weghorst, H. Gladstone, G. Raugi, D. Berg, and M. Ganter: Banded Matrix Approach to Finite Element Modelling for Soft Tissue Simulation. Virtual Reality: Research, Development and Applications, 4, 203-212.
9. H.W. Nienhuys and A.F. van der Stappen: Combining Finite Element Deformation with Cutting for Surgery Simulations. Eurographics00, Interlaken, 20-25 Aug.2000, 43-51.
10. S. Cotin, H. Delingette, N. Ayache: Real-time elastic deformations of soft tissues for surgery simulation. IEEE Trans. on Vis. and Computer Graphics, 5(1), pp. 62-73, 1999.
11. G. Picinbono, H. Delingette, N. Ayache: A. Real-Time Large Displacement Elasticity for Surgery Simulation. In Proc. MICCAI 2000, USA, October 11-14, 2000, pp. 643 – 652.
12. G. Picinbono, H. Delingette, N. Ayache: Non-linear and anisotropic elastic soft tissue models for medical simulation. Proc. IEEE Int. Conf. on Robotics and Automation (ICRA), Seoul – Korea, 2001, pp. 1370-1375.
13. W.M. Lai, D. Rubin, E. Krempl: Continuum Mechanics. Oxford: Butterworth, 1993.
14. K. Lyvan: Étude de matériaux élastiques définis par des barres. Application à la simulation en chirurgie endoscopique. PhD Thesis, Toulouse University (France), Sept.2003.
15. L.R.G. Treloar: The Mechanics of Rubber Elasticity. J. Polymer Sci.: Polymer Symposium no. 48, pp.107-123, 1974.
16. M. Mooney: A large theory of a large elastic deformation. J. Appl. Phys., Vol. 11, pp. 582-592, 1940.
17. Y.C. Fung: Elasticity of soft tissues in simple elongation. American Journal of Physiology, 213, pp. 1532-1544, 1967.
18. H. Demiray:A note on the elasticity of soft biological tissues.J.Biomechanics,5, pp.309-311.
19. P.J. Davies, F.J. Carter, A. Cuschieri : Mathematical Modelling for Keyhole Surgery Simulation: a Biomechanical Model for Spleen Tissue. IMA Journal of Applied Mathematics, Vol. 67, 1, pp.41-67, Feb. 2002.
20. R.W. Ogden: Non-Linear Elastic Deformation. Dover, 1984.
21. G. Irving, J. Teran and R. Fedkiw: Invertible Finite Elements For Robust Simulation of Large Deformation. In Proc. Eurographics / ACM SIGGRAPH Symposium on Computer Animation, pp.131-140, 2004.

Appendix

C* tensor, function of *L

The deformation gradient $\boldsymbol{F}$ being constant over $T_{\_k}$ (*P1* approximation), the relation between the edge vectors in deformed and initial states is: $\forall i, 1 \le i \le 6 : ve_i = \boldsymbol{F} \times VE_i$. On a tetrahedron, the six edge vectors can be expressed out of any three of them, e.g. matrix ***me***. Hence the six square lengths of edges can be written as a bilinear function of ***me***; using the relation to $\boldsymbol{F}$ and initial state mentioned, these square lengths can be written as a linear function of $\boldsymbol{F}^T\boldsymbol{F}$, and of the three first edge vectors in initial state (constant matrix ***ME***). The relation can be turned out to obtain the matrix $\boldsymbol{C} = \boldsymbol{F}^T\boldsymbol{F}$, as a linear function of the edges square lengths, whose constant parameters (6 (3x3) matrices $\boldsymbol{C_i}$)) depend on initial state: $\boldsymbol{C} = \boldsymbol{C}(L) = \sum_{i=1}^{6} \boldsymbol{C_i} \times l_i^2$.

Derivatives of *L*

From the relation between the edges square lengths (l_i^2) expressed with respect to the matrix ***me***, one can express their derivatives with respect to the edge vectors, and then with respect to their four defining vertices. This leads to a formulation of the derivative of *l* over each vertex $X_{0\text{-}3}$ of the tetrahedron, linear in the matrix ***me***, with 4 (6x3) constant matrices $\boldsymbol{DLM_j}$ that depend on the initial state: $0 \le j \le 3, \frac{\partial l}{\partial X_j} = \boldsymbol{DLM}_j \times \boldsymbol{me}^T$

Invariants and derivatives with respect to *L*

We denote *VTr* and ***MTr*** the 6-vector and 6x6-matrix of Traces of the matrices $\boldsymbol{C_i}$, defined by the relations on their elements: $VTr_i = Tr(\boldsymbol{C_i})$, $\boldsymbol{MTr_{i,j}} = Tr(\boldsymbol{C_i C_j})$. The three invariants of tensor *C* ($C_I = Tr(\boldsymbol{C})$, $C_{II} = 1/2\left[(Tr(\boldsymbol{C}))^2 - Tr(\boldsymbol{C}^2)\right]$, $C_{III} = \det(\boldsymbol{C}) = v^2/V^2$), can be expressed with respect to the vector of edge square lengths *l*, which is also true for the Trace of $\boldsymbol{E}$ and $\boldsymbol{E}^2$ (easily expressed in relation with Δl). Their derivatives can be expressed with respect to *l* (resp. Δl) and the matrix ***me*** - for the derivative of invariant C_{III} there is a simple expression with vectorial products of edges.