Improvement of a Topological-Physical Model to manage different physical simulations

K. Golec, M. Coquet, F. Zara, G. Damiand

Université de Lyon, CNRS
Université Lyon 1,
LIRIS, UMR5205,
F-69622, Villeurbanne, France

firstname.name@liris.cnrs.fr

ABSTRACT

We present an improvement of a unified topological-physical model which permits topological modifications during physical simulation of soft tissues. Our improvement makes the model more generic, efficient and simpler to update. The main principle of our improvement is to associate information to elements of the model, depending on the underlying physical model. Our modification of the architecture enables to easily integrate different physical models. Moreover, topological operations and physical simulations can be factorized between the different physical models. Our solution is more efficient as it leads to simpler modification algorithms after topological alterations with less changes to apply. In this paper, we present our new solution and illustrate its new properties thanks to several experiments performed on two well-known physical models: Mass-Spring System and Tensor-Mass model. The results present a comparison of our solution with the previous one for the cutting topological operation. Moreover, as our model permits to easily compare several physical models, we performed some simulations to reproduce experiments made on real tissues.

Keywords

Physical Simulation; Mass-Spring System; Tensor-Mass Model; Combinatorial Maps.

1 INTRODUCTION

In medical applications, the actual challenge is to propose physical models which provide interactive simulations with topological modifications (such as cutting). The goal is to simulate surgical operations for training. In computer graphics, the field of soft bodies simulation covers many methods [NMK+06]. The two main approaches, namely Finite Element Method (FEM) and Mass-Spring System (MSS), focus either on accuracy or on the performance of the system with visually satisfactory results. MSS is faster and easier to implement than FEM, but provides less accurate results. However, some papers focus to improve the MSS model by integrating some mechanical properties (such as Young's modulus and Poisson's ratio) in the computation of the springs properties [BSSH04, LSH07, BBJ+09].

Furthermore, as an alternative to the classic FEM, the Tensor-Mass (TM) approach has been introduced by Delingette, Cotin and Picinbono [DCA99] and extended by Schwartz [SDR+05]. This approach is based on another way to solve the mechanical equations of the objects, with a direct computation of the forces applied on each node considering the evaluation of the strain energy density. This approach allows to compute elements' information regardless of the neighborhood of this element.

In this context, this paper proposes an extension of the work of Flechon [FZDJ13, FZDJ14] who proposed a unified topological-physical model (called LCC+MSS) adapted for a mass-spring simulation that handles cutting or piercing simulated objects. Only one model is used to define the topology, the geometry and the physical properties of the simulated object, enabling topological modifications during its simulation by updating on the fly. In this paper, our contributions are:

- The improvement of the efficiency of the topological-physical model for cutting by changing the architecture of the model.
- The improvement of the topological-physical model to manage several physical models. We illustrate this issue by integrating the mass-spring system and tensor-mass model which both naturally allow dynamical topological modifications thanks to a local formulation of the strain.

- The use of the topological-physical model to simulate realistic and accurate experiments on soft tissues.

The paper is organized as follows. Section 2 presents the two physical models used. Section 3 presents the unified topological-physical model proposed by Flechon. Section 4 presents our extension of this unified model with discussion in Section 4.3. Section 5 presents some experiments showing that the performances of the LCC+MSS model have been improved by the new architecture of the topological-physical model. Moreover, Section 6 shows that our model can be used to conduce real experiments made on soft tissues and to compare physical models.

2 PHYSICAL MODELS

Mass-Spring System. In computer graphics, mass-spring systems have largely been used to simulate the behavior of deformable objects [TPF91, BFA02]. This physical model is based on the discretization of the object into a set of masses (also called particles) interconnected by springs. Then at each time step, the following main steps are performed for each particle:

(1) Computation of the forces applied on the particle due to springs and external forces. Remember that the force involved by a spring connecting particles i and j, with stiffness constant k_{ij} and initial length l_{ij}, is defined at time t by $F_{ij}(t) = F^e_{ij}(t) + F^v_{ij}(t)$.

- $F^e_{ij}(t)$ is the elasticity force of this spring with

$$\begin{cases} F^e_{ij}(t) &= k_{ij}\left(\|P_j(t) - P_i(t)\| - l_{ij}\right) u_{ij}(t), \\ F^e_{ji}(t) &= -F^e_{ij}(t), \end{cases}$$

  where $P_i(t), P_j(t)$ are the positions of i, j and $u_{ij}(t) = (P_j(t) - P_i(t))/\|P_j(t) - P_i(t)\|$.

- $F^v_{ij}(t)$ is the viscosity force of this spring with

$$F^v_{ij}(t) = \gamma_{ij}\left(V_j(t) - V_i(t)\right) \cdot u_{ij}(t)\, u_{ij}(t),$$

  where $V_i(t), V_j(t)$ are the velocities of i, j and $\gamma_{ij} = 2\sqrt{k_{ij}(m_i + m_j)/2}$ the spring's viscosity coefficient where m_i, m_j are the masses of i, j.

(2) Computation of the acceleration according to Newton's second law with $\frac{d^2}{dt^2}P_i(t) = F_i(t)/m_i$ where $F_i(t)$ represents the forces applied on particle i.

(3) Computation of the velocity and position according to the acceleration using a numerical integration scheme (for instance the semi-implicit Euler one).

Tensor-Mass model. The TM approach is based on the discretization of the object into several elements as for the FEM, but then the equations are solved locally, making this approach more suitable for topological modifications of the object. To account for various mechanical behaviours, several formulations have been presented: linear Hooke's model [CDA00], non-linear geometrical model based on Saint Venant-Kirchhoff's elasticity model [PDA00], anisotropic material [Pic03] and non-linear visco-elastic deformations [SDR$^+$05] with some pre-computations to accelerate the process.

The simulation's loop of the MT approach involves the following main steps for each element E of the object:

(1) Computation of the displacement of a point X inside E with

$$U_E(X) \simeq \sum_{j=0}^{n-1} \Lambda_j(X)\, U_j,$$

where n is the number of 3D nodes P_j defining the element, U_j the displacement of each node P_j from its initial position, and $\Lambda_j(X)$ some interpolation functions defined according to the type of element used for the discretization.

(2) Computation of the deformation energy according to the displacement of the element's nodes. In our work, we consider a non-linear elasticity behavior using the Green-Saint Venant strain-tensor

$$\varepsilon(X) = \frac{1}{2}\left(\nabla U^T(X) + \nabla U(X) + \nabla U^T(X)\, \nabla U(X)\right).$$

The associated strain energy density is defined by

$$W(X) = \frac{\lambda}{2}\left(\text{tr}\ \varepsilon(X)\right)^2 + \mu\ \text{tr}\ \varepsilon(X)^2,$$

where λ and μ are the Lamé coefficients characterizing material stiffness.

(3) Computation of the elasticity force on any node $P_j \in E$ for $j \in [0, n-1]$ with

$$F_E(P_j) = -\frac{\partial W_E(P_j)}{\partial U_j},$$

where $W_E(P_j)$ is the energy density of deformation of the considered element evaluated at node P_j.

(4) Computation of the acceleration according to Newton's second law (in the same way that for MSS).

(5) Computation of the deformation and displacement of the object using a numerical integration scheme (in the same way that for MSS).

In this paper, we used the approach proposed by Faure [FZJM12]. The formulation of the forces applied on each node of the elements was generated thanks to symbolic computation. Thus, the steps (1), (2) and (3) of the simulation loop were replaced by the direct formulation of the force applied on the nodes.

3 LCC+MSS MODEL

3D Linear Cell Complex. The unified topological-physical model (called LCC+MSS) proposed by Flechon [FZDJ13, FZDJ14] used as underlying data structure the 3D Linear Cell Complex (LCC) [Dam12] from the CGAL Open Source geometric algorithms library [The12]. With this topological structure, an orientable object is represented using 3D combinatorial maps (called 3-maps) [Lie94, DL14] as a subdivision of 0-cells (vertices), 1-cells (edges), 2-cells (faces) and 3-cells (volumes). The cells are described by darts, a generalization of half-edges [Män87] to higher dimension, plus pointers between these darts called β_i with $i \in \{0,1,2,3\}$. The 3D linear cell complex data structure is then obtained by associating to 0-cells the coordinates in $\mathbb{R}^3$ of corresponding points.

Fig. 1 represents two adjacent cubes described by a 3-map. Darts are displayed as arrows. $\beta_1(d)$ is the next dart following dart d in the same face and the same volume. $\beta_0(d)$ is the previous dart in the same face and volume. $\beta_2(d)$ gives the other dart from d belonging to the same edge, the same volume but not the same face. $\beta_3(d)$ gives the other dart from d belonging to the same edge, the same face but not the same volume.

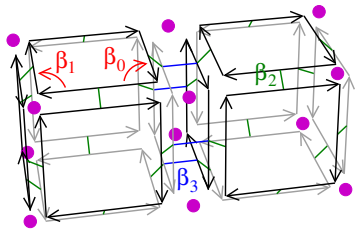

Figure 1: Two adjacent cubes described by a 3-map. Each cube has 6×4 darts. Purple disks represent vertices associated with points to obtain a 3D LCC.

This representation using darts and pointers describes the cells of an object as well as all the incidence and adjacency relationships. If two i-cells share a common $(i-1)$-cell, they are adjacent (for example two volumes sharing a common face). If one cell belongs to the boundary of another cell, these two cells are incident (for example a vertex belonging to the boundary of a face). This information is very useful to allow topological modifications during the simulation.

Lastly, any information can be associated with cells thanks to the *attribute notion* (*e.g.* a length to edges or a color to faces). We denote by *i-attribute(d)* the attribute associated with the i-cell containing dart d.

3D LCC for MSS. Flechon added information to a 3D LCC to construct the LCC+MSS model which is suitable for a MSS based on the Baudet formulation for hexahedral mesh [BBJ+09]. This formulation is available for basic hexahedra including four inner springs.

In the LCC+MSS model, a data structure `Particle` is associated to each 0-cell thanks to a 0-attribute and a data structure `Spring` is associated to each 1-cell thanks to a 1-attribute. Four additional springs inside each hexahedron are also described by the `Spring` data structure. But contrary to particles and springs previously described, these inner springs are not associated with any cells of the 3-map, but directly associated to the object.

- The `Particle` structure stores information related to a particle: its mass, velocity, acceleration and the sum of the forces applied to it. Moreover, it stores the list of the inner springs attached to it.
- The `Spring` structure stores the physical properties of a spring: its initial length, its stiffness and its two extremities (thanks to two pointers to the two `Particle` connected by the considered spring).

Fig. 2 illustrates an example of a LCC+MSS model for two adjacent cubes.

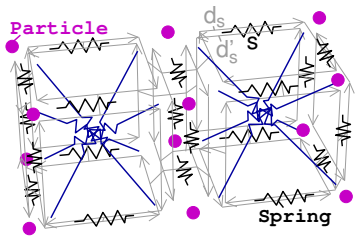

Figure 2: Two adjacent cubes described by the LCC-MSS model which is a 3D LCC associated with particles (purple disks), springs (in black) and inner springs (in blue).

Interest of using topological model for physical simulation. The classical data-structure used in MSS simulations is a graph where vertices correspond to particles and edges correspond to springs. This simple model allows to directly and easily compute the forces resulting from the springs and to accumulate these forces on the particles. However, the treatment becomes more difficult for topological modifications. Indeed, it is not possible with such a graph, to directly retrieve the elements of the object, nor all the neighborhood relations between the elements, the springs and the particles when these relations are necessary to implement topological modifications.

For example, to cut an object between two adjacent hexahedra (*i.e.* two hexahedra sharing a face), particles of the face could need to be duplicated (as we will see later with Fig. 4). To detect if a particle needs to be duplicated or not, we need to search for existence of a path of adjacent elements joining the two cut hexahedra. This is not possible directly in the graph. Thus, a data-structure has to be added to represent elements by

storing indices of its particles and indices of its adjacent elements. Then, we also need to add to each particle the indices of its incident elements. These additional data-structure and links need to be initialized and updated coherently after each operation. Moreover, other data-structures and links will be required to implement another topological modification.

The main interest of using a topological model is that we are sure that all the topological relations are described and can be retrieved directly. Existing topological constraints ensure the topological validity of the described objects and can be used to test the consistency of implemented operations. Moreover, we can use many existing and proved topological operations that can serve as basic tool to implement high level operation in our physical simulation.

4 OUR NEW SOLUTION

In this paper, we propose an improvement of the LCC+MSS model. At this time, we consider only hexahedral elements due to the physical models used, but the structure is suitable for any topology. Our model follows two new principles: (1) to store the darts of each element in 3-cells of the 3-map; (2) to associate the physical information directly in the corresponding cells of the 3-map.

4.1 Storing of the darts in their element

For the first principle, we add in each element an array called `dart` containing all the darts of the element grouped by faces *i.e.* an array `dart[6][4]` is associated to each hexahedron. The initialization of these arrays is performed at the construction of the 3-map using the convention given in Fig. 3. With this structure, all points of a given hexahedron can be retrieved thanks to the stored darts. For example, point X_0 is obtained from `dart[0][3]` and point X_1 is obtained from `dart[4][2]`. Moreover, each point of an hexahedron can be retrieved by three different darts. For example X_0 can also be retrieved thanks to `dart[1][0]` and `dart[4][1]`.

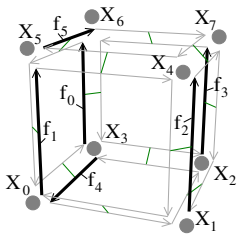

Figure 3: Convention used to store darts of a hexahedron composed of points X_i with $i \in \{0,\dots,7\}$. $f_j =$ `dart[j][0]` with $j \in \{0,\dots,5\}$ correspond to the first darts of each face (the bolded arrows).

One major interest of our proposal is to provide a direct access to all the incidence relations between cells, allowing a direct access in $O(1)$ to information required by the different physical models. This direct access is done either by using the relations given by the 3-map or by the relations given by the `dart` array.

4.2 Storing of physical information in the corresponding cells

The second principle is based on the fact that the physical information associated to the *i*-cells (with $i \in \{0,1,2,3\}$) of the 3-map is depending on the physical model chosen. Consequently, the direct association of the physical information to the corresponding cells allows to consider easily different physical models. In this subsection, we present the use of our proposal for a mass-spring system and a tensor-mass model, but any physical model can be envisaged. We terminate this subsection by showing the generic computation of the forces accumulation thanks to this second principle. Naturally, the computation of the acceleration of the object and its integration in order to obtain its velocity and its position is the same for all physical objects who are based on the newtonian dynamics.

Mass-Spring System. For a mass-spring system based on the Baudet formulation [BBJ+09], particles are associated to 0-cells thanks to 0-attributes, springs are associated to 1-cells thanks to 1-attributes, and inner springs are associated to 3-cells thanks to 3-attributes. Indeed, we remember that in the considered MSS, each hexahedron includes 4 inner springs s_i with $i \in \{0,1,2,3\}$. By convention, we fix spring s_0 (*resp.* s_1, s_2, s_3) between particles (P_0,P_7) (*resp.* (P_0,P_7), (P_1,P_6), (P_2,P_5) and (P_3,P_4)) where particles P_i correspond to point X_i with $i \in \{0,\dots,7\}$ presented in Fig. 3.

To compute the forces involved by springs (supported both by 1-cells and 3-cells), each spring needs to know its two extremity particles which are the parameters of the `addForce()` function used for this purpose. This can be retrieved thanks to the 3-map and to the `dart` array, without storing explicitly the link between springs and particles.

- For each spring s associated to a 1-cell, one dart d_s of the edge is directly known thanks to the 3-map (see Fig. 2). The first particle extremity of the spring is given by 0-attribute(d_s). The second particle extremity of the spring is given by 0-attribute(d'_s) with $d'_s =$`other_extremity(`d_s`)` the dart associated with the second extremity of the edge.
- For any inner spring s_i associated to a 3-cell, its two particles extremity are directly retrieved by our convention and the `dart` array. For example for s_0, its first extremity is 0-attribute(`dart[1][0]`) and its second extremity is 0-attribute(`dart[3][1]`).

 ISBN 978-80-86943-65-7

Tensor-Mass model. For the tensor-mass model, nodes are associated to 0-cells thanks to 0-attributes, and the formulation of the forces applied on each node of an element are associated to 3-cells thanks to 3-attributes. In this paper, we only consider hexahedral mesh with a non-linear behavior using the Green-Saint Venant strain-tensor, but it could easily be extended to any topology or physical behavior thanks to the formulation of the forces proposed by Faure [FZJM12].

To compute the forces applied on each element inside the `addForce()` function, each element starts by computing the displacement of their nodes using their current and initial position. Thanks to the `dart` array, each node of a 3-cell is retrieved in constant time. Then, the force applied on each node of an element is directly computed using the formulation of forces generated by symbolic computation.

Generic forces accumulation. Thanks to this new way to associate physical information to a LCC, the forces applied on the object are easily computed by performing a loop on all the relevant i-cells, that is the i-cells affected by the force computation (*e.g.* the 1-cells and 3-cells for MSS; the 3-cells for MT) and calling the appropriate `addForce()` function on the associated physical information. In practice, it consists to iterate through all the enabled i-attributes (*i.e.* the attributes which are associated to cells of the 3-map).

4.3 Comparison with previous solution

As recalled in Section 3, the initial definition of the LCC+MSS topological-physical model [FZDJ13] associates physical entities to cells (similarly than in our new solution), except for inner springs which are not associated with elements. Moreover, each spring stores its two extremity particles to be able to accumulate the force of this spring to the particles.

Improvement for cutting. With this previous solution, extremity particles of springs may have to be updated after a cutting. For example in Fig. 4, the cut of the two top hexahedra leads to the split of particles v_1 and v_2 into four particles v_1, v_1', v_2, v_2'. But after this cutting, springs of the top-right hexahedra are still associated with original particles v_1 and v_2 instead of particles v_1' and v_2' (see Fig. 4(a)). To solve this problem after the cutting, a post-processing is applied to update the incorrect extremities of springs (see Fig. 4(b)).

The solution proposed [FZDJ13] is to update all the springs having as extremity the particles involved in the cutting. This requires to store in each particle the list of the springs touching it. Then, when a particle is duplicated due to a cutting, an iteration is performed through this list to update the extremities of the springs to point to the new particles. Moreover, the list of springs is split in two parts: the first part for springs incident to v_1 and the second one for springs incident to v_1' (the new particle). This treatment requires a complex and time consuming processing.

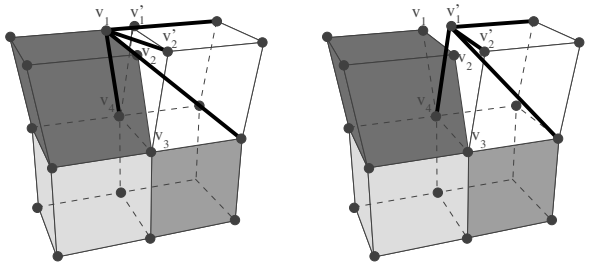

Figure 4: In the LCC+MSS model (pictures from [FZDJ13]), the cutting involves the updating of the spring extremities. (a) The bold segments represent the inner springs wrongly attached to particle v_1 after the cutting. (b) The same springs updated after a post-processing step.

With our new solution, as the darts of an element are constant whatever the topological modification applied on its neighbor elements, no change is required for cutting. For example, the darts of the two hexahedra split in Fig. 4 are still valid after the cutting. Moreover, as springs and inner springs extremities are only defined by using the darts of the hexahedra, they are still valid after the cutting without any updating. Thus our algorithms with topological modifications are simpler than Flechon and our method is faster (as shown in our experiments in Section 5) than Flechon.

Improvement for any physical models. Another advantage of our proposal concerns the ease to consider several physical models as presented in previous subsection. Indeed, as physical information are associated with cells thanks to the 3-map, the same algorithm is used to accumulate forces regardless of the physical model used (in our actual implementation, for both mass-spring system and tensor mass model).

Moreover, the 3-map and the `dart` array are up to date after cutting without any need of post-processing. Thus, only physical properties concerning by the cutting (like the mass) have to be updated. This simplifies the integration of new physical models in our system.

5 RESULTS

In this section, we present results obtained in a Intel®i5 4690 CPU, 4 cores @3.50 GHz with 16 Go RAM. For the following tests, we consider a gravity put to $g = -9.8$ m.s^{-2} and beams with a density $\rho = 1000$ Kg.m^{-3}, a Young's modulus $E = 400$ KPa and a Poisson's ratio $\nu = 0.3$.

Time for cutting. We start by a comparison in time for cutting between our approach and that of

Flechon [FZDJ13]. In Fig. 5, the curves represent the time in secondes involved to cut a beam compared to the number of faces disconnected. We consider beams discretized in $200 \times 200 \times 2$, $400 \times 400 \times 2$, $600 \times 600 \times 2$ and $800 \times 800 \times 2$ elements of size $1 \times 1 \times 1$ m with a cutting according to the Z axis.

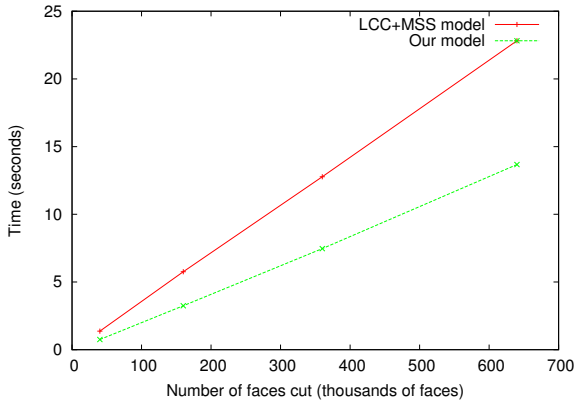

Figure 5: Comparison for the time involved for cutting.

In Table 1, more complex meshes are used for the comparison in time (in ms) for cutting: a frog (2,354 elements), an elephant (125,754 elements), a mushroom (27,000 elements) and a giraffe (85,405 elements).

Mesh	# faces cut	Flechon	Our model
Frog	112	4.0	2.0
Elephant	592	14.5	7.6
Mushroom	987	32.2	17.7
Giraffe	5210	176.3	91.6

Table 1: Comparison of time (in ms) for cutting.

We observe that our method is nearby twice as fast than the LCC+MSS model as less information has to be updated in our approach to take into account a topological modification of the object.

Precision. Fig. 6 presents a comparison for the traction test between the results obtained using the MSS and the MT approach compared to the analytic solution with beams discretized in $5 \times 5 \times 20$ and $10 \times 10 \times 40$ elements. We consider the analytic solution given by

$$y = \frac{\rho g l h}{24\,E\,I}(4Lx^3 - 6L^2x^2 - x^4)$$

with $L = 1$ m the length, $h = 0.25$ m the height, $l = 0.25$ m the thickness of the beams and $I = lh^3/12$ the inertia moment. Fig. 7 presents a visual comparison of the equilibrium state obtained with the $10 \times 10 \times 40$ MSS beam and the $10 \times 10 \times 40$ MT beam.

As expected, we observe that the MT simulation is closest to the analytic solution than MSS simulation. This experiment shows the interest of having several physical models in a same software to simply comparisons between different simulations.

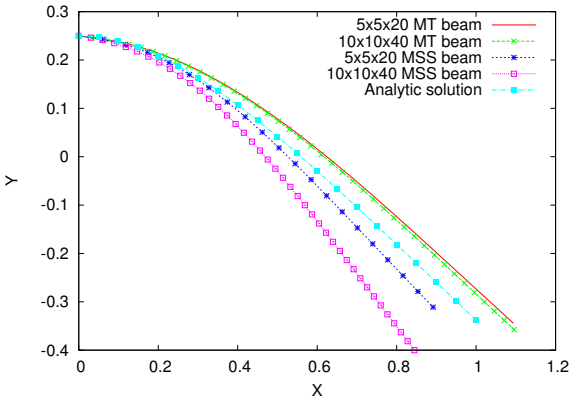

Figure 6: Comparison between the MSS, the MT and the analytic solution for a traction test.

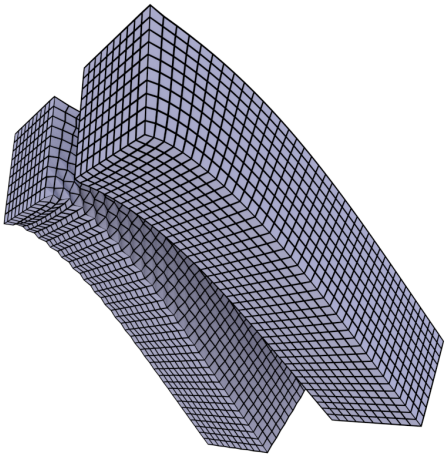

Figure 7: Comparison between the MSS (left) and the MT (right) for a traction test performed on beams discretized in $10 \times 10 \times 40$ elements.

Time of execution. Fig. 8 presents a comparison in time for the traction test performed on beams with several discretization. The aim was to vary the number of elements of size $0.025 \times 0.025 \times 0.025$ m. We present sequential and parallel time with a parallelization of the simulation's loop performed using the Intel®TBB (Threading Building Blocks) C++ template library for task parallelism. Four threads were involved for the parallel time. We obtained an average speedup of 1.45 for the MSS and 2.73 for the MT.

The MT simulation is slower than the MSS simulation due to the more complex physical equations. However its parallelization has a better speedup than the MSS one. This is due to the fact that MT uses only once type of attribute for the physical information (associated with 3-cells), while MSS uses two different types of attributes (1-cells for springs and 3-cells for inner springs) requiring two different loops.

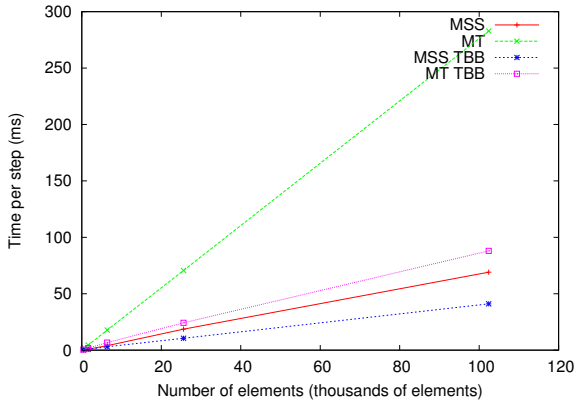

Figure 8: Comparison in time for the MSS and MT simulation performed in sequential or in parallel with TBB.

Illustrations of cutting. Fig. 9 and 10 show cutting made during a MT or a MSS simulation of meshes subject to the gravity. These pictures show that our model allows topological modifications of the objects during their physical simulation.

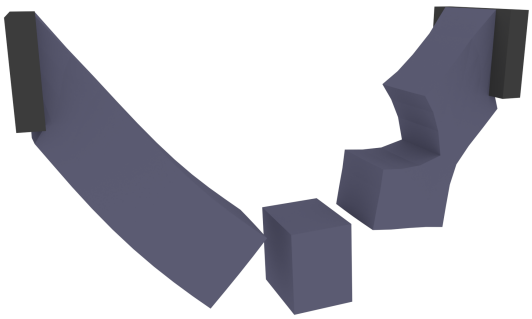

Figure 9: Cutting made during a MT simulation of a beam (subject to gravity) fixed to its two extremities.

Figure 10: Cutting made during a MSS simulation. Some particles were fixed and the gravity was applied. The density of the letters was put to $\rho = 200$ Kg.m^{-3}.

Fig. 11 shows several cuts applied on a giraffe mesh (with no simulation) that contains 85,405 elements. The different parts were separated by hand to obtain a proper visualization.

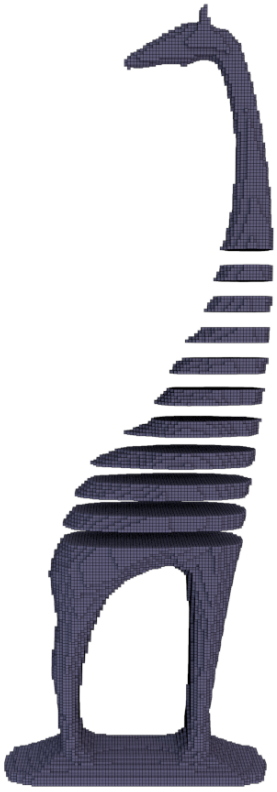

Figure 11: Ten successive cuts on a giraffe mesh.

6 REPRODUCE REAL EXPERIMENTS

In this section, we present the use of our model to simulate experiments performed on real tissues. Firstly, we explain the original experiments made on porcine tissues [NNP12]. Secondly we present our simulation and the preliminary results obtained with the MSS and the MT model. Note that the comparison between two physical models in this simulation is possible thanks to our improvement of the physical-topological model.

Experiment. The original experiments were performed on a porcine liver disks of $2-3.7$ mm thickness and diameter of approximately 15 mm. Fig. 12 presents a schematic picture of the experiment. The disks of porcine tissues were glued between the plates of a rotational rheometer. Then, the top plate was rotating at a frequency varied from 0.1 Hz to 2 Hz, while the bottom one was constrained. This experiment is equivalent to a local deformation of the porcine disks.

Figure 12: Schematic representation of the experiments made on porcine tissues [NNP12].

 ISBN 978-80-86943-65-7

The authors [NNP12] of the original experiments derived a non-linear model from the results obtained with the experiments. However, since we only consider the linear case, the following equations are modified accordingly.

The model is represented by the relaxation modulus $G(t)$ and the stress-strain relationship $\sigma[\varepsilon(t)]$. We have:

$$G(t) = \frac{K}{\Gamma(1-n)} t^{-n}$$

with K the consistency, n the linear constitutive index and t the time [KM09, NVP10]. The Gamma function is defined by

$$\Gamma(x) = \int_0^\infty t^{x-1} e^{-t} dt.$$

Considering the experiments made on liver tissues, we used as values $K = 861$ Pa.s^n and $n = 0.1138$ [NNP12]. The stress-strain relationship is defined for the linear case by

$$\sigma[\varepsilon(t)] = \int_0^t G(t-t')\dot{\varepsilon}(t')dt',$$

where σ represents the stress, $\varepsilon(t)$ the strain and $\dot{\varepsilon}$ a constant strain rate. We considered the same three strain rates as the authors [NNP12] with $\dot{\varepsilon} \in \{0.0151\ s^{-1}, 0.133\ s^{-1}, 0.67\ s^{-1}\}$.

Simulation. We used our model to reproduce the experiments by simulation. We assumed that locally the tissue is represented by a rectangular block of infinitely long plates. The bottom of the tissue did not move and the top of the tissue only moved on the abscissa X. For this, we applied a direct displacement to the top particles and studied the response of the tissue. The simulated tissue consists of 3,751 particles with $10 \times 10 \times 30$ volumes (see Fig. 13). The unit of a cube is 1 mm.

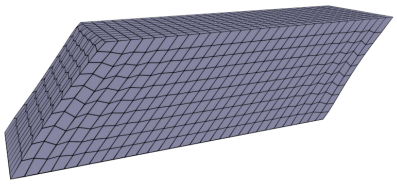

Figure 13: MSS composed of $10 \times 10 \times 30$ hexahedra to reproduce the experiments made on tissues.

As in the LCC+MSS model of Flechon, the formulation of Baudet [BBJ+09] is used to compute the stiffness constant according to Poisson's ratio and Young's modulus to integrate physical parameters into the MSS.

Results. We considered porcine liver which is a non-compressible tissue with $E = 1000$ Pa, $\nu = 0.499$ and $\rho = 1000$ Kg/m^3. The gravity force is set to zero as it is negligible in the real experiment. As we performed a shear experiment in the linear case, the strain is small enough to be negligible. For this reason, we consider the stress-time relationship to analyse our simulations.

Fig. 14 shows the stress according the time obtained for the MT model and the MSS model. For the simulations, the stress is computed by dividing the force applied on the object by the area of the top plate of the tested tissue (in our case 300 mm^2).

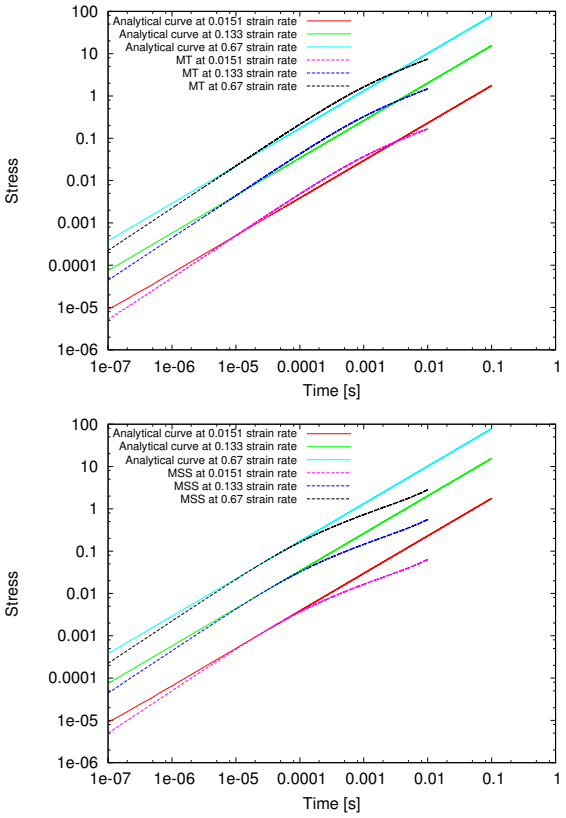

Figure 14: Evolution of stress in time for respectively the MT model (top) and the MSS model (bottom) for three different strain rates.

As expected, we observe that MT and MSS have a linear behavior for small deformations.

In the future work, we will improve the MSS solutions by implementing volume preservation and applying some correction forces. We also want to implement a new damping solution to ensure the correctness of the model for linear case.

7 CONCLUSION AND PERSPECTIVE

In this paper, we presented an improvement of the LCC+MSS model proposed by Flechon [FZDJ13]. Our improvement is based on two new principles: (1) to store the darts of each element in the 3-cells of the 3-map; (2) to associate the physical information directly in the corresponding cells of the 3-map.

Thanks to these two new principles, our model allows simpler algorithms since no more modifications are required for topological modification operations. Secondly, our model gives better results since less operations are required. Lastly, our model is more generic since it is now easier to add a new physical model while keeping all the existing operations. We have illustrated these improvements in our experiments with better results obtained for cutting than the previous model. Moreover, two different physical models (namely the mass-spring system and the tensor-mass model) have been presented who are both based on the same unified topological model. With these two physical models, we have illustrated the generic computation of the accumulation of the forces thanks to our two new principles.

In future work, we are considering adding others physical models to easily compare the results of simulations. The most interesting would be to implement the Finite Element Method and give thought to the Position Based Dynamics approach, like the one of Müller [MHHR07]. According to the MSS, we plan to include the volume preservation constraint in order to improve its precision. We plan to study the work of Aubert [AB97] which created a space deformation model (called DOGME), to deal with the volume preservation, and the more recent article of Duan [DHC+14] who treats about preserving volume by position corrections. Furthermore, to improve the precision of our simulations, it is also important to take a deeper look into constraining the particles to distribute correctly the acting forces. Lastly, we want to extend our simulation of soft tissues to the case of non-linear mechanical behavior to consider bigger deformations. All these future work will benefit of our new solution of the LCC+MSS model allowing to define simpler and more efficient algorithms to perform topological modifications during simulation for any kind of physical objects.

Acknowledgments

This work was supported by the LABEX PRIMES (ANR-11-LABX-0063) of Université de Lyon, within the program "Investissements d'Avenir" (ANR-11-IDEX-0007) operated by the French National Research Agency (ANR). The authors would like to thank the following people for their support: E. Flechon for providing the original version of TopoSim (software implementing the LCC+MSS model), X. Faure and F. Jaillet for providing the code to compute the forces generated in the mass-tensor model.

8 REFERENCES

[AB97] F. Aubert and D. Bechmann. Volume-preserving Space Deformation. *Computers & Graphics*, 21(5):625 – 639, 1997.

[BBJ+09] V. Baudet, M. Beuve, F. Jaillet, B. Shariat, and F. Zara. Integrating Tensile Parameters in Hexahedral Mass-Spring System for Simulation. In *WSCG'2009*, 2009.

[BFA02] R. Bridson, R. Fedkiw, and J. Anderson. Robust treatment of collisions, contact and friction for cloth animation. *ACM Trans. Graph.*, 21(3):594–603, July 2002.

[BSSH04] G. Bianchi, B. Solenthaler, G. Székely, and M. Harders. Simultaneous Topology and Stiffness Identification for Mass-Spring Models based on FEM Reference Deformations. In *MICCAI 2004*, volume 2, pages 293–301. Springer, November 2004.

[CDA00] S. Cotin, H. Delingette, and N. Ayache. A Hybrid Elastic Model allowing Real-Time Cutting, Deformations and Force-Feedback for Surgery Training and Simulation. *Visual Computer*, 16(8):437–452, 2000.

[Dam12] G. Damiand. Linear cell complex. In *CGAL User and Reference Manual*. CGAL Editorial Board, 4.0 edition, 2012.

[DCA99] H. Delingette, S. Cotin, and N. Ayache. A Hybrid Elastic Model Allowing Real-Time Cutting Deformations and Force Feedback for Surgery Training and Simulation. In *Computer Animation'99*, pages 70–81, États-Unis, 1999.

[DHC+14] Y. Duan, W. Huang, H. Chang, W. Chen, J. Zhou, S. K. Teo, Y. Su, C. Chui, and S. Chang. Volume Preserved Mass-spring Model with Novel Constraints for Soft Tissue Deformation. *IEEE journal of biomedical and health informatics*, 2194(c):1–12, November 2014.

[DL14] G. Damiand and P. Lienhardt. *Combinatorial Maps: Efficient Data Structures for Computer Graphics and Image Processing*. A K Peters/CRC Press, September 2014.

[FZDJ13] E. Flechon, F. Zara, G. Damiand, and F. Jaillet. A generic topological framework for physical simulation. In *WSCG'2013*, pages 104–113, June 2013.

[FZDJ14] E. Flechon, F. Zara, G. Damiand, and F. Jaillet. A unified topological-physical model for adaptive refinement . In *VRIPHYS'2014*, pages 39–48, September

2014.

[FZJM12] X. Faure, F. Zara, F. Jaillet, and J.-M. Moreau. An Implicit Tensor-Mass solver on the GPU for soft bodies simulation. In *VRIPHYS'2012*, pages 1–10, December 2012.

[KM09] J. F. Kelly and R. J. McGough. Fractal ladder models and power law wave equations. *The Journal of the Acoustical Society of America*, 126(4):2072–81, October 2009.

[Lie94] P. Lienhardt. N-dimensional generalized combinatorial maps and cellular quasi-manifolds. *Int. J. Comput. Geometry Appl.*, 4(3):275–324, 1994.

[LSH07] B. A. Lloyd, G. Székely, and M. Harders. Identification of spring parameters for deformable object simulation. *IEEE Transactions on Visualization and Computer Graphics*, 13(1):1081–1093, 2007.

[Män87] M. Mäntylä. *An Introduction to Solid Modeling*. Computer Science Press, Inc., New York, NY, USA, 1987.

[MHHR07] Matthias Müller, Bruno Heidelberger, Marcus Hennix, and John Ratcliff. Position based dynamics. *J. Vis. Comun. Image Represent.*, 18(2):109–118, April 2007.

[NMK+06] A. Nealen, M. Müller, R. Keiser, E. Boxerman, and M. Carlson. Physically Based Deformable Models in Computer Graphics. *Computer Graphics Forum*, 25(4):809–836, December 2006.

[NNP12] S. Nicolle, L. Noguer, and J.-F. Palierne. Shear mechanical properties of the spleen: experiment and analytical modelling. *Journal of the mechanical behavior of biomedical materials*, 9:130–6, May 2012.

[NVP10] S Nicolle, P Vezin, and J-F Palierne. A strain-hardening bi-power law for the nonlinear behaviour of biological soft tissues. *Journal of biomechanics*, 43(5):927–32, March 2010.

[PDA00] G. Picinbono, H. Delingette, and N. Ayache. Real-Time Large Displacement Elasticity for Surgery Simulation: Nonlinear Tensor-Mass Model. In *Proceedings of MICCAI'00*, pages 643–652, London, UK, 2000. Springer-Verlag.

[Pic03] G Picinbono. Non-linear anisotropic elasticity for real-time surgery simulation. *Graphical Models*, 65(5):305–321, September 2003.

[SDR+05] J.-M. Schwartz, M. Denninger, D. Rancourt, C. Moisan, and D. Laurendeau. Modelling liver tissue properties using a non-linear visco-elastic model for surgery simulation. *Medical image analysis*, 9:103–112, 2005.

[The12] The CGAL Project. *CGAL User and Reference Manual*. CGAL Editorial Board, 4.0 edition, 2012.

[TPF91] D. Terzopoulos, J. Platt, and K. Fleischer. Heating and melting deformable models. *The Journal of Visualization and Computer Animation*, 2(2):68–73, 1991.