

Elastons: Implementation Notes

Sebastian Martin*
ETH Zurich

Peter Kaufmann†
ETH Zurich

Mario Botsch‡
Bielefeld University

Eitan Grinspun§
Columbia University

Markus Gross¶
ETH Zurich

Abstract

This supplementary document explains in more detail the implementation of the plasticity model as well as the virtual node and cutting algorithms mentioned in the main paper.

1 Introduction

Objects can not only transition between solid, shell, and rod in space, regime changes can also develop over time. In plastic and viscous deformations, for instance, material can be stretched into thin sheets or strands, whose elastic behavior can correctly be captured by our approach. The topological changes induced by cutting and fracturing can also generate material with small features in all three dimensions that should be captured by the simulation system.

2 Plasticity

In order to show the applicability of our approach to these scenarios we adapt the additive plasticity model given in [O'Brien et al. 2002] to our approach and define plastic membrane and bending strain variables α_p, β_p^k at the elastons. By taking the difference between the measured geometric strains α_g, β_g^k (corresponding to equations (5) and (6) in the main paper) and the stored plastic strains, we can define the effective elastic strain as

$$\alpha_e = \alpha_g - \alpha_p, \quad \beta_e^k = \beta_g^k - \beta_p^k, \quad (1)$$

Then the new elastic energy stored in a single elaston becomes

$$W_{\text{elaston}} = \frac{V}{2} \left(\alpha_e : \mathbf{C} : \alpha_e + \sum_{k=1}^3 \frac{(h_k^e)^2}{12} \beta_e^k : \mathbf{C} : \beta_e^k \right). \quad (2)$$

Additionally to the elastic forces, the gradient of the energy with respect to the DOFs now also contains the elaston's contribution to the plastic force

$$\mathbf{f}_i^p = \mathbf{A}_i^T \mathbf{C} \alpha_p + \sum_k (\mathbf{B}_i^k)^T \mathbf{C} \beta_p^k. \quad (3)$$

This force can simply be incorporated into the right-hand side of the equation of motion

$$\rho \ddot{\mathbf{u}} - \frac{\partial W}{\partial \mathbf{u}} = \mathbf{f}.$$

For updating the plastic strains we rely on the same criteria as proposed in [O'Brien et al. 2002] and adapt it straightforwardly to our strain representation.

While this simple model already allows for the simulation of some plastic deformation, it is not suitable for larger deformations when combined with a linear strain measure. We therefore perform a periodic resampling step similar to the approach of [Wojtan and Turk

2008] such that the plastic deformation is represented by the geometry and does not need to be represented by the plastic strain variables anymore. Similar to [Müller et al. 2004], we do however not transfer the entire plastic strain into the rest state geometry which would remove any remaining elastic component, but perform the resampling by the three steps discussed next.

Rest State Update In order to get a continuous transition of the material position without having to worry about popping artifacts we take the current configuration as the new rest state, which will guarantee C^0 -continuity of the solution across the resampling, similar to [Wojtan and Turk 2008]. Material points, GMLS samples, and elastons can all three be generated from scratch for the new rest state as described in the sampling section of the main paper. Since the new rest state entirely describes the current positional state of the simulation, the new displacement field \mathbf{u} can be set to zero.

Notation We will always use superscripts n for referring to quantities computed in the new rest configuration. As in the main paper, we use $\bar{\mathbf{x}}$ to denote positions in the old rest state and \mathbf{x} for the deformed configuration which is identical to the new rest state.

Velocity Field Update In order to also have a smooth transition of the motion across the resampling, i.e., approximate C^1 -continuity in time, we find the new velocity field $\mathbf{v}^n(\mathbf{x})$ by minimizing the least squares error between the new and the old velocity field, defined as

$$\int_{\Omega^n} \|\mathbf{v}^n(\mathbf{x}) - \mathbf{v}(\bar{\mathbf{x}}(\mathbf{x}))\|^2 d\Omega^n. \quad (4)$$

Note that these two fields are defined on separate domains. However, these domains are mapped one-to-one by $\mathbf{x}(\bar{\mathbf{x}}) = \bar{\mathbf{x}} + \mathbf{u}(\bar{\mathbf{x}})$. Since the two fields are represented in their corresponding bases as $\mathbf{v}^n(\mathbf{x}) = \sum_i \mathbf{v}_i^n N_i^n(\mathbf{x})$ and $\mathbf{v}(\bar{\mathbf{x}}) = \sum_i \mathbf{v}_i N_i(\bar{\mathbf{x}})$, the least squares problem basically becomes the following linear system:

$$\mathbf{G}^n \mathbf{v}^n = \mathbf{G} \mathbf{v} \quad (5)$$

where $\mathbf{G}_{ij}^n = \int_{\Omega} N_i^n N_j^n$ is the Gram matrix of the new basis, and $\mathbf{G}_{ij} = \int_{\Omega^n} N_i^n N_j$ is a matrix transferring the old velocity field into the new configuration. In order to evaluate these integrals numerically, we do again take advantage of the higher-order integration scheme used for the elaston derivation and mass matrix computation and perform this integration up to second order.

Plasticity Update The only remaining step is the transfer of the plastic variables from the old to the new rest state. In order to understand the reasoning for the following procedure, we quickly recapitulate the three states involved (depicted in Fig. 1). First, we have the undeformed configuration $\bar{\mathbf{x}}$ in which the discretization variables live. Second, we have a (virtual) physical rest state, which is described implicitly by the plastic strain variables α_p, β_p^k (dashed region), and third, we have the current configuration \mathbf{x} described by the actual displacement field as $\mathbf{x} = \bar{\mathbf{x}} + \mathbf{u}(\bar{\mathbf{x}})$. During resampling, we update the undeformed configuration to the current state. The current configuration then coincides with the new undeformed configuration. However, the physical rest state, which can change due to plastic deformation, should stay the same during the resampling

*e-mail: smartin@inf.ethz.ch

†e-mail: peterkau@inf.ethz.ch

‡e-mail: botsch@techfak.uni-bielefeld.de

§e-mail: eitan@cs.columbia.edu

¶e-mail: grossm@inf.ethz.ch

process. This can be achieved by defining the new plastic strain as the negative of the old elastic strain [Müller et al. 2004].

This idea is realized in two steps. First, the new plastic strain can be defined in the old rest configuration. This is done by evaluating the total membrane and bending strains contained in the deformation field $\mathbf{u}(\bar{\mathbf{x}})$. Note that since we are using a corotational approach, actually the rotated deformation field at the elaston's position is taken for the strain evaluation. Then (1) can just be used to define the new plastic strains in the undeformed configuration as $-(\alpha_e + \sum_k \theta_k \beta_e^k)$. Note that this strain is still living in the old undeformed configuration. Given a direction \mathbf{d} in the old configuration, the stretch in this direction can be measured as $-\mathbf{d}^T (\alpha_e + \sum_k \theta_k \beta_e^k) \mathbf{d}$, where θ is the evaluation point in coordinates of the elaston's local frame.

In a second step, in order to use this updated plastic strain in the new rest state, it has to be transformed to the new configuration (Fig. 2). We assume the local frames of two elastons in the old and new rest state configurations are given as columns of the 3×3 matrices \mathbf{T} and \mathbf{T}^n . Then we can use the deformation gradient $\mathbf{F} = \mathbf{I} + \nabla \mathbf{u}$ to define a matrix \mathbf{M} that transforms a material direction \mathbf{d}^n in the local frame of the deformed configuration to a direction \mathbf{d} in the local frame of the undeformed configuration as

$$\mathbf{M} = \mathbf{T}^T \mathbf{F}^{-1} \mathbf{T}^n. \quad (6)$$

Using this map, we can compute stretch along \mathbf{d}^n in the new rest state by mapping it back and evaluating it in the undeformed configuration using the old strain. This gives us

$$(\mathbf{d}^n)^T (\alpha_p^n + \sum_k \theta_k \tilde{\beta}_p^{kn}) \mathbf{d}^n = -(\mathbf{d}^n)^T \mathbf{M}^T (\alpha_e + \sum_k \theta_k \beta_e^k) \mathbf{M} \mathbf{d}^n, \quad (7)$$

leading to corresponding plastic strains in the new rest state as

$$\alpha_p^n = -\mathbf{M}^T \alpha_e \mathbf{M} \quad (8)$$

and

$$\tilde{\beta}_p^{kn} = -\mathbf{M}^T \beta_e^k \mathbf{M}. \quad (9)$$

While these two strains now allow for measuring stretch in directions defined in the new rest state, there are still two things missing.

First, the strain in the vicinity of the elaston's center is measured as $\alpha_p^n + \sum_k \theta_k \beta_p^{kn}$ where θ describes the evaluation point in the elaston's local frame. However, in the definition given above, θ is still given in the old rest state's local frame and needs also to be transformed with the mapping \mathbf{M} as

$$\theta = \mathbf{M} \theta^n. \quad (10)$$

Incorporating this transformation into the new bending strains gives then

$$\beta_p^{kn} = \sum_j \mathbf{M}_{jk} \tilde{\beta}_p^{jn}, \quad (11)$$

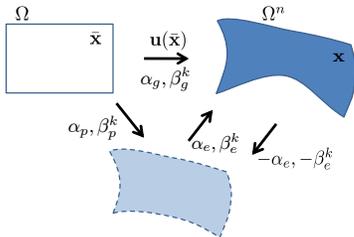


Figure 1: Illustration of geometric, elastic and plastic strain, as well as the new plastic strain after resampling.

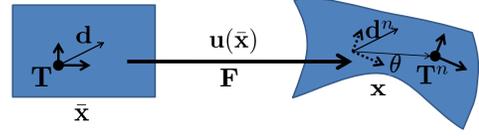


Figure 2: Illustration of an elaston's local frame in the rest state, its mapped position in the new rest state, as well as the frame of a nearby elaston whose plastic strain is extrapolated during resampling.

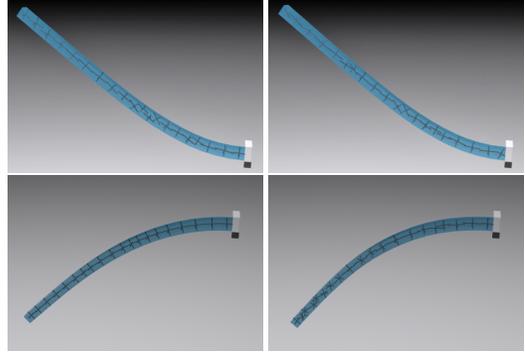


Figure 3: Example of dynamic resampling of a swinging bar showing elaston configurations before (left) and after (right) resampling. Thanks to the smooth position and velocity interpolation, the simulation is free of any visible popping artifacts. The full animation can be seen in the accompanying supplementary video.

such that the strain at a location θ^n in the new rest state can finally be evaluated as

$$\alpha_p^n + \sum_k \theta_k \beta_p^{kn}. \quad (12)$$

So far, we are able to transform the plastic strain from a position in the undeformed configuration to the corresponding position in the deformed configuration. However, we would like to sample the new rest state independently from the old sampling. In a first step, we therefore sample the new positions together with the new local frames. For each of these new elastons we then look up the closest transformed elaston and use its new coordinate frame for transferring the plastic strain to the transformed elaston.

As a final step, we then just need to extrapolate these strains to the new elastons. We do this by defining a coordinate frame at the closest transformed elaston where the axes correspond to the local frame of the new elaston. In this frame, we can express the elaston's position as θ and since we assume the strain to be a linear function in the vicinity of the transformed elaston, we can simply extrapolate the plastic membrane strain to the new position as

$$\alpha_p^n + \sum_k \theta_k \beta_p^{kn}. \quad (13)$$

The bending strain stays the same. While different interpolation schemes could be chosen for interpolating the plastic strain to the new elaston locations, we did not observe any problem using this linear extrapolation approach.

3 Cutting and Meshless Virtual Node

Cutting applications are a further area where thin structures can appear dynamically and which require a proper handling by the simulation method. For cutting simulations, we need to introduce a notion of connectivity between elastons. We do this by first computing a connectivity graph between the densely sampled candidate points \mathcal{M} , where two candidate points are connected if they overlap. The Lloyd relaxation performed in the elaston sampling step associates each candidate point with an elaston, which also implies a connectivity graph between elastons (see Fig. 4 for an example). Cutting can then be implemented by simply removing connections between two previously connected candidate points.

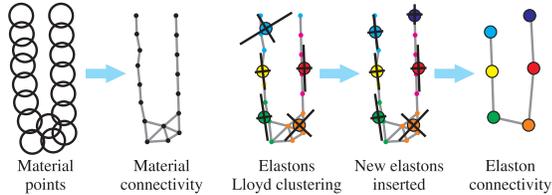


Figure 4: *Overlapping material spheres implicitly imply connectivity (left). Elastons containing disconnected material components are split up and elaston connectivity is computed (right).*

Before starting the simulation, and after each change to the connectivity graphs, a meshless equivalent of the “virtual node algorithm” of [Molino et al. 2004] is performed. This consists of two steps.

First, new elastons are introduced to make sure the local connectivity graph of each candidate subset \mathcal{M}_i is in itself connected. This can be performed by looking at the material subset \mathcal{M}_i of an elaston and considering the connections between these points. If this local connectivity graph consists of multiple disconnected components, the old elaston is removed and new elastons are created, one for each component. The local frame of the new elastons is again found through a covariance analysis.

Second, for each GMLS sample point, the local connectivity graph of all elastons influenced by that sample point is considered. If this connectivity graph consists of multiple disconnected components, the GMLS sample point is duplicated and each component gets its own independent copy (see Fig. 5).

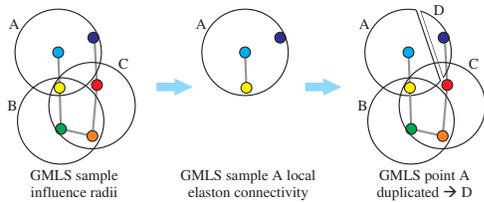


Figure 5: *DOFs containing disconnected elastons in their support are duplicated for each component.*

References

- MOLINO, N., BAO, Z., AND FEDKIW, R. 2004. A virtual node algorithm for changing mesh topology during simulation. *ACM Trans. on Graphics* 23, 3, 385–392.
- MÜLLER, M., KEISER, R., NEALEN, A., PAULY, M., GROSS, M., AND ALEXA, M. 2004. Point-based animation of elastic,

plastic and melting objects. In *Proc. of Symp. on Computer Animation*, 141–151.

- O’BRIEN, J. F., BARGTEIL, A. W., AND HODGINS, J. K. 2002. Graphical modeling and animation of ductile fracture. *ACM Trans. on Graphics* 21, 3, 291–294.

- WOJTAN, C., AND TURK, G. 2008. Fast viscoelastic behavior with thin features. *ACM Trans. on Graphics* 27, 3, 47:1–47:8.