Check for updates

DOI: 10.1111/cgf.70382

EUROGRAPHICS 2026 / B. Masia and J. Thies
(Guest Editors)

COMPUTER GRAPHICS *forum*

# Affinification: A Fine Approximation of Deformations

A. Mercier-Aubin[1,4], T. Schneider[2], P.G. Kry[3], and S. Andrews[4]

[1]Université de Sherbrooke, Canada
[2]University of Victoria, Canada
[3]McGill University, Canada
[4]École de Technologie Supérieure, Canada

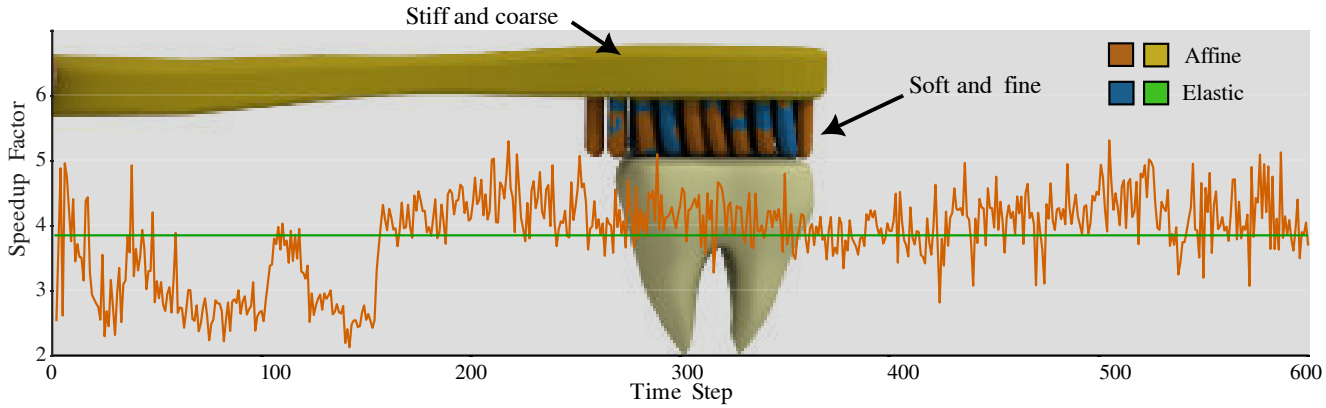


**Figure 1:** *We fully simulate a toothbrush with heterogeneous materials. The bristles are soft while the base is stiff. Affinification provides steady improvements in performances compared to traditional simulation methods as it allows bristles to contain large coarsened regions. The green line shows the mean speedup of 3.85×, while the orange plot line shows the speedup per time step.*

**Abstract**

*We introduce affinification, a novel method for accelerating physics-based animation of elastic solids. During a time-dependent simulation, our method automatically partitions the space into affine and elastic regions depending on the deformation. As such, we capture localized deformations while significantly reducing computational costs with larger regions of model reduction. We design a new clustering method based on deformation rates to capture affinely deforming regions, and explore multiple heuristics for seeding, pattern generation, and the impact of physical parameters on coarsened regions. We compare our method with the ground truth, showing performance increasing with resolution and recorded simulations up to 17× faster compared to elastic simulations, while retaining similar levels of visual fidelity.*

**Keywords:** Collisions, Physics-based, Soft-Bodies, Affine Bodies, Contact, Adaptive, Coarsening, Subspace

## 1. Introduction

Simulating deformable bodies efficiently is a longstanding challenge in physics-based animation. Many real-time applications, such as video games, virtual reality, and surgical simulations, require fast and stable computations while preserving the realistic behaviour of chosen materials. However, simulating large, highly detailed deformable objects at full resolution is computationally expensive. Physics-based animation has long grappled with the trade-off between simulation accuracy and computational efficiency. Recent simulation algorithms rely on GPU pipelines to enhance performance. However, in many practical scenarios, such as mobile platforms or applications where the GPU is heavily utilized, access to the GPU may be limited or unavailable. This highlights the need for novel and efficient techniques that can effectively leverage the CPU [LLL*25]. As such, we provide a cheap, linear complexity oracle, coupled with a system implemented on CPU for time integration.

© 2026 The Author(s). Computer Graphics Forum published by Eurographics - The European Association for Computer Graphics and John Wiley & Sons Ltd.
This is an open access article under the terms of the Creative Commons Attribution-NonCommercial License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited and is not used for commercial purposes.

Fine-scale deformations demand substantial computational resources, requiring simulations to either predefine resolution choices or adopt adaptive approaches to dynamically balance speed and accuracy (e.g., $h$ and $p$ refinement). However, most of these methods focus on refinement (i.e., carefully increasing the cost to improve accuracy) as derefinement requires coarsening a mesh during simulation and thus has to transfer the solution across domains with different boundaries [NSO12; NSPO14; SRH*16]. Few methods exist [FSKP23], but they keep the boundary fixed, and the cost of remeshing often outweighs the benefits.

An alternative strategy for improving performance is runtime adaptive model reduction, where the fidelity of the physical model is adaptively refined (or approximated) in response to changes during simulation. In this spirit, we introduce *affinification*, a novel reduction technique based on affine bodies that effectively approximates regions of an elastic body undergoing scaling, shearing, and non-uniform deformation, in addition to rigid motion. This approach enables expressive deformations in coarsened regions, allowing plausible representation of material behaviour while reducing computational cost. Our key insight is that by permitting affine transformations of reduced regions, we retain many of the computational benefits of standard model reduction, and even improve them as it allows more aggressive selection of regions to include actively affinely deforming elements. The main challenge is to efficiently identify affine and elastic regions of simulation bodies. We evaluate affinification across a suite of simulations and demonstrate that it consistently outperforms both fully elastic simulations and relevant adaptive methods in terms of computational efficiency, while maintaining high visual fidelity. Our method achieves recorded speedups ranging from 2× to over 16× compared to traditional soft body simulations depending on model resolution, with greater benefits observed in higher-resolution scenes due to the complexity of traditional solvers. It remains stable under conditions such as soft-body impacts, heterogeneous materials, and dynamic contact scenarios. Ablation studies confirm that deformation-rate-based clustering and randomized seeding strategies yield more coherent and efficient affine regions than the other heuristics tested.

While affinification and elastification share some structural similarities, our novel solution to these challenges has significant differences from the techniques used for rigidification. Affinification allows generalized deformation, particularly in regions undergoing shearing or stretching. To do so, we designed a different full-space to affine mapping based on Kronecker products and propose a projection to affine space. We introduce a greedy clustering method using a breadth-first search that groups elements based on affine velocity similarity, using a dynamic center of deformation bounds. Additionally, we present a novel resource limiting feature that enables performance management for resource constrained applications, like mobile applications, by restricting the number of elastic elements per frame. Such approach is reasonable for affinification, but not for rigidification due to the large artifacts generated.

## 2. Related Work

In physics-based animation, generating the fine details of deformation is costly, requiring the simulation to choose between accuracy and speed of animation. Typically, the discretization of models [DDCB01], or sparsity of control points [GKS02] allows the user to tune this trade-off to their needs before even running the simulation. However, because these choices are made a priori and are too strictly enforced, they can miss important details that other approaches, like adaptive methods, would detect.

Static condensation [BC96] reduces models to only the surface vertices. Later, subspace condensation [HTC*14; TMDK15] dynamically adjusts the simulation basis to capture large-scale motion efficiently, and reduces artifacts compared to static condensation. While their method focuses on modal subspaces [BJ05], our affine clustering can be seen as a complementary strategy that adapts the simulation model itself rather than its basis. Subspace methods often struggle with large rotations and require careful basis construction. In contrast, our method operates directly in full space and adapts the simulation model itself by clustering elements into affine bodies, enabling expressive deformation without precomputed bases.

Progressive Cloth Simulation [ZDF*22] introduces a quasistatic framework that enriches coarse meshes with fine-scale energies to produce predictive previews consistent with high-resolution models. By refining solutions while guaranteeing feasibility through C-IPC contact and strain-limit handling, this method achieves high-fidelity drapes without altering material behavior, contrasting with our approach that instead allows trivially tuneable accuracy.

Tangentially, autoencoder based reduction [SYS*21; FMD*19], can efficiently generate fast simulations with small reduced space, at the cost of requiring data, and the need to train a model. We prefer avoiding this step and provide a simulator that works without the need to gather information prior to the execution.

Alternatively, freezing methods [AR12; MFRC13] remove degrees of freedom in low-motion regions to accelerate simulation but do not support internal deformation. Similarly, it is possible to rely on hierarchies for adaptive refinement [NKJF09; TNFG14]. While these approaches operate in the inertial frame and are primarily suited for particle systems, they share conceptual similarities with our goal of reducing computational effort in minimally deforming regions. Affinification generalizes these ideas by allowing coarsened regions to undergo affine motion, capturing a broader range of behaviours while maintaining computational efficiency.

Using rigid body dynamics or motions can be useful to speed up elastic simulations [BZA14]. Inspired by adaptive merging [CBK20], adaptive rigidification of solids [MKWL22] and shells [MK23] both use an oracle to detect when rigidified parts can become elastic again. These methods can be seen as a generalization of freezing and a form of model reduction without the need for a hierarchy, which is accomplished by adaptively coarsening and refining the scene using rigid body patterns as mesh resolutions. However, the coarsened region potentially misses important motions that should accumulate over time, like a steady push constantly below the elastification threshold. In a similar fashion, Trusty et al. [TFLK24] adaptively pick subspaces through the use of an oracle to find the most adequate subspace for a time step. This approach requires updates and downdates to the subspaces, a steady monitoring of error, and a potentially expensive oracle with large dense subspace Hessians compared to the linear oracle

© 2026 Eurographics - The European Association for Computer Graphics and John Wiley & Sons Ltd.

14678659, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1111/cgf.70382 by Ecole De Technologie Superieur, Wiley Online Library on [12/06/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

**Algorithm 1:** Main loop

Find contacts
$\Delta\dot{\boldsymbol{x}}_{\text{approx}} \leftarrow$ Predict velocities // §3.4
$\dot{F}_{\text{approx}} = B(\dot{\boldsymbol{x}} + \Delta\dot{\boldsymbol{x}}_{\text{approx}})$ // Eq. 4
BFS($\dot{F}_{\text{approx}}$, seeds) to find affine components // Alg. 2
$\dot{\boldsymbol{q}}_t \leftarrow$ Project velocities onto affine space // Eq. 9
$\Delta\dot{\boldsymbol{q}} \leftarrow$ reduced space solve ($\dot{\boldsymbol{q}}_t$) // Eq. 10
$\Delta\dot{\boldsymbol{x}} \leftarrow$ Map changes in velocities to full space // Eq. 12
$\Delta\dot{\boldsymbol{x}}_c \leftarrow$ PGS contact solve ($\Delta\dot{\boldsymbol{x}}$) // §3.1
$\dot{\boldsymbol{x}}_{t+1} \leftarrow \dot{\boldsymbol{x}}_t + \Delta\dot{\boldsymbol{x}}_c + \Delta\dot{\boldsymbol{x}}$
$\boldsymbol{x}_{t+1} \leftarrow \boldsymbol{x}_t + h\dot{\boldsymbol{x}}_{t+1}$

of adaptive rigidification. These techniques share key similarities with our approach, including the use of an oracle and runtime model reduction. However, like rigidification, we aim to provide a method suitable for resource-constrained systems and so focus our comparisons there.

Adaptive methods like rigidification can also take the form of a hierarchical [Mül08; WZQ*17] multigrid approach that is layer-based, rather than geometric or algebraic [MK24], to accurately solve for motions. However, this is slower than the performance of approximate physical models as it requires a fully elastic layer to remain accurate. We draw inspiration from this work for our resource-limiting feature by using partially ordered sets of elements to determine what should remain affine under strict computational budgets.

A different physical representation for rigid bodies consists of relaxing the rigid body constraint to allow affine deformations [LKL*22]. Such a system becomes more malleable, avoiding numerical issues on contacts and likewise permitting efficient linear continuous collision detection as opposed to needing to deal with rigid body rotation. We combine the strengths of affine body dynamics and adaptive rigidification approaches [Mer25] to benefit from enhanced performance of an adaptively coarsened model while accumulating important dynamic deformations of the coarsened regions with affine transformations.

## 3. Methodology

The proposed method, which we call *affinification*, incorporates affine bodies as an intermediate representation between fully elastic and rigid states. Affinification introduces deformations such as scaling and shearing, enabling a flexible approximation of deformation while maintaining computational efficiency. As shown in Algorithm 1, our method starts with a discretized mesh that can be used for elastic simulations. We then predict the velocities of the elements that were turned affine in previous time steps. Next, we find new clusters of affinely deforming elements and form affine bodies. We then map all the degrees of freedom (DOFs) to the reduced space and solve the reduced system. We reintegrate the velocities into the full space. The following section will describe each part of the main loop through the subsections.

### 3.1. Time Integration

While the reduction method that we will present is implementation-agnostic, we provide implementation details for reproducibility purposes. We use a linearized backward Euler [BW98] time integrator

$$A = M - hD - h^2K, \tag{1}$$

$$\boldsymbol{b} = h(\boldsymbol{f} + D\dot{\boldsymbol{x}}_t + hK\dot{\boldsymbol{x}}_t), \tag{2}$$

where $K$ is the stiffness matrix, $M$ the mass matrix, $D$ a Rayleigh damping matrix, $h$ the time step size, $t$ is the time step, and $\boldsymbol{f}$ are forces. The system we solve for the fully elastic simulation in the comparisons of the result section is therefore $A\Delta\dot{\boldsymbol{x}} = \boldsymbol{b}$.

To resolve contacts, we use a projected Gauss-Seidel (PGS) [Erl04] solver to compute frictional contact impulses. The contact Jacobian is constructed from signed distance queries, and warm-started impulses from the previous frame are used to accelerate convergence.

### 3.2. Clustering

An important step in the affinification process involves identifying regions within a deformable object that can be simplified without significantly altering the overall behaviour of the simulation. The method evaluates how the deformation gradient

$$F = B\boldsymbol{x}, \tag{3}$$

changes over time to determine whether a region should be classified as elastic or affine. Strain rate provides insight into how much an element deforms relative to its previous shape, while the deformation gradient rates,

$$\dot{F} = B\dot{\boldsymbol{x}}, \tag{4}$$

capture local changes in transformations beyond simple translation and rotation. We assemble the deformation rate with the same kinematic mapping $B$ as the deformation gradient, but with a multiplication over velocities. We replace the velocities $\dot{\boldsymbol{x}}$ for the evaluation of previously affine elements with their approximation from the prediction to obtain approximate deformation rates. Deformations rates are particularly well-suited for affine clustering as they are inhenrently affine transformations in tetrahedral models. It is reasonable to expect elements with similar deformation rates, to affinitely deform in similar ways. We explain the approximation process in Subsection 3.4. By integrating the new measure, the classification mechanism better adapts to a wider range of affine motions.

Once we compute the deformation rates, the next step is to form affine bodies by clustering elements that exhibit similar affine velocities as shown in Algorithm 2. This grouping process relies on detecting regions where connected elements share similar deformation trajectories, ensuring a coherent approximation of local material behaviour. The optimal affine velocity for each cluster is computed to best represent the combined motion and momentum of its elements, incorporating the change in translation, rotation, scaling, and shearing components. Rather than looking for the best cluster, we find a greedy clustering for fast linear-time assembly of

© 2026 Eurographics - The European Association for Computer Graphics and John Wiley & Sons Ltd.

14678659, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1111/cgf.70382 by Ecole De Technologie Superieur, Wiley Online Library on [12/06/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

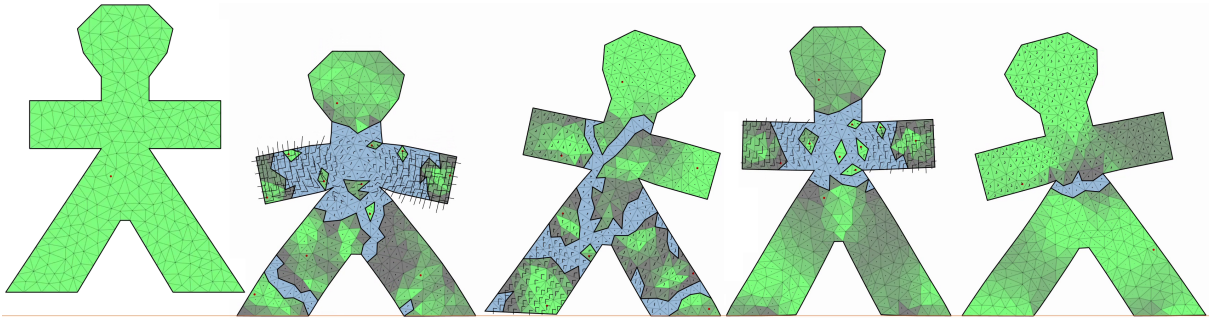

**Figure 2:** *The per element frames show the deformation rates $\dot{F}$ columns for a stick man animation evolving through time, where the character is in constant movement, yet features large chunks of affine regions. The elements spanning from green to black show their $\dot{F}$ distance to the $\dot{F}_{centre}$, blue elements are elastic, and red dots are the seeds for each affine cluster.*

affine bodies. Figure 2 illustrates an example of affine clustering in a 2D elastic simulation.

Our clustering uses the $L_2$ norm of the deformation gradient rate $\dot{F}$ to the centre of the deformation rate bounds $\dot{F}_{\text{centre}}$ for a cluster. Initially, the bounds of a new cluster are set to the deformation rate of the first element. On merge, we adjust the minimum $\dot{F}_{\text{min}}$ and maximum $\dot{F}_{\text{max}}$ bounds with a component-wise min and max operation on the 9 elements of the deformation gradient rate. We then adjust the centre of bounds

$$\dot{F}_{\text{centre}} = 0.5(\dot{F}_{\text{min}} + \dot{F}_{\text{max}}) \tag{5}$$

to be exactly at the center of $\dot{F}_{\text{min}}$ and $\dot{F}_{\text{max}}$. We add an element to the cluster when the deformation rate distance

$$d = \|\dot{F}_{\text{centre}} - \dot{F}\| \tag{6}$$

from the centre of bounds to the element to merge is below the threshold $\tau$. When clusters include only a single element, we discard them because keeping them would yield the same number of degrees of freedom without any added benefit. Like rigidification, we discard elements of clusters that are connected to other clusters by a single vertex (hinge) to simplify the implementation. It would be possible to handle these cases usings constraints to joint clusters, but it is unclear if the extra constraint computation would be worth the few elements discarded.

We use the threshold for pairs of elements, the current element to visit in the breadth-first search (BFS) and the neighbour to merge. We have two thresholds set globally; the more restrictive $\tau_A$ to determine if elements should turn affine and the more permissive $\tau_E$ to determine if elements should stay affine. Because thresholds change between elastic elements and those already affine, this can cause potential inconsistencies in the bounds based on the order of nodal exploration. This means that the clustering process can be order dependent on insertion of elastic elements, which could lead to inconsistencies in cluster formation and introduce temporal instability (e.g., flickering between affine and elastic states). To reduce the impact of the traversal order on the distance threshold, we look at previous states of both elements. So the merge threshold $\tau$ varies between the two globally set thresholds depending on if it is currently affine or elastic. Hence, if the element visited or the neighbour to merge are elastic, then we use the affinification thresholds $\tau_A$, else we use the elastification threshold $\tau_E$.

**Algorithm 2:** Identify new affine bodies by greedy connected component BFS.

```
Data: search start element e, affine body index j to assign.
if e visited then return
Enqueue e in Q
Ḟ_min ← e.Ḟ, Ḟ_max ← e.Ḟ, Ḟ_centre ← e.Ḟ
while Q is not empty do
    Dequeue e from Q
    Set e to visited
    if any vertex of e has a hinge then continue
    Assign body j to element e
    Ḟ_min ← min(e.Ḟ, Ḟ_min)        // Component-wise min/max
    Ḟ_max ← max(e.Ḟ, Ḟ_max)
    Ḟ_centre ← 0.5(Ḟ_min + Ḟ_max)                    // Eq. 5
    for vertex v in element e do
        Assign body j to vertex v
    end
    for neighbour n of element e do
        if n visited then continue
        // Threshold set according to last state
        if n or e is elastic then
            τ ← τ_A
        else
            τ ← τ_E
        end
        d ← ||Ḟ_centre − n.Ḟ||                       // Eq. 6
        if d < τ ∨ n ∈ C then // C defined in §3.5
            Enqueue n in Q
        end
    end
end
```

© 2026 Eurographics - The European Association for Computer Graphics and John Wiley & Sons Ltd.

14678659, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1111/cgf.70382 by Ecole De Technologie Superieur, Wiley Online Library on [12/06/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

### 3.3. Reduction

The affine reduced-space mapping is a key component of the algorithm done through a mapping matrix

$$G = \begin{bmatrix} I & 0 \\ 0 & \Gamma \end{bmatrix}. \tag{7}$$

The upper left part features an identity matrix for the elastic DOFs, while the lower right maps to affine bodies. The sparse $\Gamma$ matrix acts as both a selection matrix and a mapping to reduced space, as such no reordering is required. Each block of $\Gamma$ contains a submatrix

$$\Gamma_i = \begin{bmatrix} I_3 \\ I_3 \otimes \boldsymbol{r}_i \end{bmatrix}, \tag{8}$$

where the operator $\otimes$ denotes the Kronecker product, $\boldsymbol{r}_i$ are vectors from the centre of the affine body to each affinified vertex, and $I_3$ is a sized 3 identity matrix. This effectively maps the vertex elastic degrees of freedom to 12 degrees of freedom. An affinified chunk has 3 degrees of freedom for the translation and 9 degrees for a 3 by 3 affine matrix (shear, scale, rotation). The affine part of the mapping is a flattening operation on affine coordinates.

Before solving for time integration, we first compute the weighted least-squares projection to affine space

$$\dot{\boldsymbol{q}}_t = (G^T M G)^{-1} G^T M \dot{\boldsymbol{x}}_t, \tag{9}$$

to be momentum preserving. Our mapping transforms the problem into a reduced system, represented as

$$G^T A G(\Delta \dot{\boldsymbol{q}}) = G^T \boldsymbol{b}, \tag{10}$$

where $\dot{\boldsymbol{q}}$ are the projected velocities onto affine space.We also make sure to preserve the projection discrepancy,

$$\boldsymbol{d}_t = G\dot{\boldsymbol{q}}_t - \dot{\boldsymbol{x}}_t, \tag{11}$$

between full space velocities $\dot{\boldsymbol{x}}_t$ and reduced space velocities $\dot{\boldsymbol{q}}_t$ to add back into the system after the solve. Note that $G\dot{\boldsymbol{q}}_t$ maps the solved velocities back into full-space, making our discepency measure also in full-space.

The system to solve in Equation 10 has the potential to be much smaller, containing a mixture of affine and elastic parts. To solve this system, we use Matlab's LDL decomposition and direct solve. We do a single Newton iteration, as it is usually sufficient for stability. We note that like in Mercier-Aubin and Kry [MK23], it is possible to reuse the same mapping if we need more iterations. Likewise, the reduction allows faster assembly of the constraint Jacobian for contact handling as the Delassus operator for the PGS solve is applied to the reduced system.

We compute the centre of masses for affine bodies as well as the vectors to particles $\boldsymbol{r}_i$ in the soft body rest frame. This is to ensure the stiffness matrix remains consistent with the material properties even when condensed into the reduced-space. This allows the reduced-space system to still optimize for elastic deformations respecting material properties inside the affine regions. The method is therefore not limited to the energy model from Lan, Kaufman, Li, et al. [LKL*22], which constrains bodies to be as rigid as possible. Affinification works with any standard deformation energy model.

Once we have the reduced-space change in velocities $\Delta\dot{\boldsymbol{q}}$, we map it back to full space

$$\Delta\dot{\boldsymbol{x}} = G\Delta\dot{\boldsymbol{q}} + \boldsymbol{d}, \tag{12}$$

in order to integrate the particles in time for the full space model.

### 3.4. Prediction of Deformation

We need information about the elastic deformation of elements in affine state. Following Mercier-Aubin, Kry, Winter, and Levin [MKWL22], we use a single iteration of a preconditioned conjugate gradient (PCG) to compute the approximate change in velocities,

$$\Delta\dot{\boldsymbol{x}} = A^{-1}(\boldsymbol{b} - J_c^T\lambda) + h\ddot{\boldsymbol{x}}_{\boldsymbol{g}}, \tag{13}$$

which we can then use to detect what should remain affine, like in Algorithm 2. Because the system solves for changes in velocity, there is no need for a warm start. As such, the initial iterate is an array of zeros. Here $J_c^T$ is the contact Jacobian, $\lambda$ is the contact impulse, and $\ddot{\boldsymbol{x}}_{\boldsymbol{g}}$ is the acceleration due to gravity. We split gravity from the solve to reduce the load on the single iteration of PCG. We avoid assembling the full system matrix. Instead, we use a precomputed incomplete Cholesky factorization of a Laplacian-based Hessian approximation from Mercier-Aubin and Kry [MK23];

$$A_p = M + h^2 B^T W B. \tag{14}$$

This allows quick diffusion of elastic information.

Unlike rigidification, we do not require the first few frames to be fully elastic as the monitoring of changes in deformations is on a single frame basis rather than over multiple frames. The reason this was implemented in adaptive rigidification was to prevent locking at energy dips, such as when a cantilever beam reaches its lowest point of swing. However, with affinification, the affine transform still allows deformation motions, which prevents locking. As such, affinification can happen in the first simulation frame.

We include warm-started contact forces from the previous frame and approximated new contact forces as explicit external forces in the one iteration PCG solve. This ensures that contacts are captured in the prediction as demonstrated in Figure 3. For new contacts, we estimate them by solving cheap bilateral constraints.

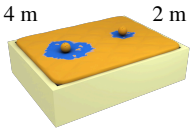


**Figure 3:** *Dropping a bowling ball from higher elastifies a larger region.*

### 3.5. Resource Limiting

To support resource-constrained applications with predictable performance, we introduce an optional resource limiting strategy that restricts the number of elastic degrees of freedom (DOFs) per frame. This addition to the contributions of this paper was implemented for both affinification and rigidification for testing purposes. In the multi-layer solver from Mercier-Aubin and Kry [MK24], multiple incremental rigid layers are built based on percentages of rigidification from a list of elements ordered by

© 2026 Eurographics - The European Association
for Computer Graphics and John Wiley & Sons Ltd.

14678659, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1111/cgf.70382 by Ecole De Technologie Superieur, Wiley Online Library on [12/06/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

strain rates. This effectively generates many nearly rigid clusters at different resolutions.

Much like a one-layer version of the multi-layer solver, we pick a fixed percentage of elements that must belong to affine clusters, but do so from deformation rates $\dot{F}$ with smallest neighbour differences, placing the corresponding elements into the set $\mathcal{C}$. This need not be a complete sort, but only to find the unordered k-smallest values. Because of this, we can use a linear complexity algorithm for k-smallest values [KKZZ19]. These elements then join clusters alongside the ones that adaptively coarsen with our method due to the normal thresholds. As the number of elastic elements is limited by our chosen percentage, we make possible resource-limiting, which is required for interactive applications. For the rigidification implementation, we instead find the k-smallest strain rates to turn rigid.

While this lumping of low deforming groups of elements to affine clusters can lead to stiffening, it generally gives reasonable clusters as neighbouring elements of low strain rate elements tend to also be of relatively low strain rate. Likewise, this reduces the variability of the time integration scheme in terms of wall clock time. We provide this for applications that require this feature, but likewise acknowledge that such resource limiting cannot fully represent the global elastic motions compared to the rest of this paper.

## 4. Results

We evaluate our method across a variety of benchmark scenes, comparing against both fully elastic simulations and adaptive rigidification.

We built this project on a fork of the open-source adaptive rigidification repository from GitHub in order to fairly compare the performances of both methods. Hence, the simulator is coded in Matlab and runs entirely on CPU. Critical parts are implemented in mex C++ for all the methods, like the PCG and PGS solvers or the clustering algorithms. We ran the experiments on a consumer grade laptop with an Intel® Core™ Ultra 7 155H Processor. We likewise used the original adaptive rigidification examples as a benchmark to compare the performance of the methods as well as new examples.

### 4.1. Ablation Studies

We conducted ablation tests to evaluate the impact of coarsening patterns and clustering seed selection. In Figure 4, we compare random patterns to patterns made from the deformation rate distances. The random patterns are generated by merging neighbouring elements with 80% success chance. We notice that the random patterns generate many tiny clusters that get removed automatically. Likewise, the random patterns do not generate coherent affine clusters, potentially stiffening affine chunks with conflicting elastic elements. As such, it is better to use the deformation rate clusters because they tend to generate larger clusters with reasonable affine motions. This is because the deformation rates are inherently affine transformations.

In Figure 5 we compare the seed where we start forming the clusters. The position of the seed affects the clusters as we use a

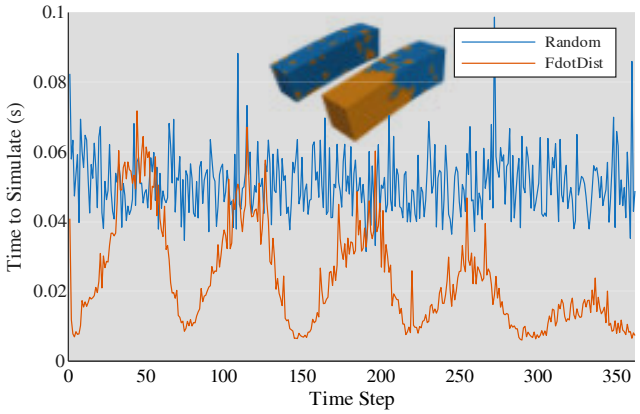


**Figure 4:** *Randomly selecting patterns leads to poor performance compared to deformation based clustering.*

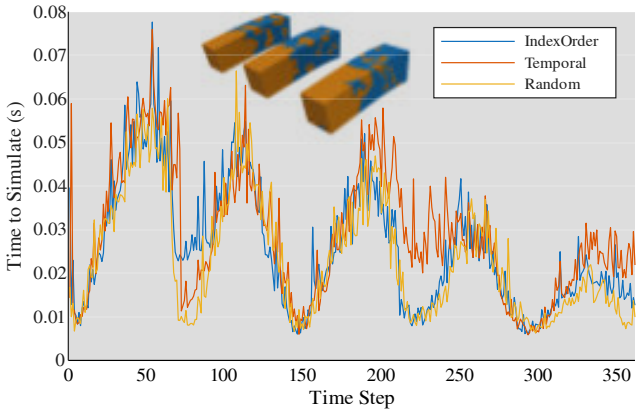


**Figure 5:** *Random seeding outperforms temporal seeding as it has more opportunities to find good seeds, which leads to better clusters.*

greedy clustering algorithm. As such, starting with a given seed changes the initial bounds to satisfy for merging two elements. Temporal coherence is not necessary, but can help to see how regions evolve with less hectic changes in reduced regions. We nudge the algorithm in the direction of the previous cluster to achieve coherence by preserving the cluster seeds from one time step to the next. We add new seeds greedily to cover empty regions and skip seeds when they already belong to a cluster. Alternatively, we can use index-based seeding that starts the BFS seeds in element ID order, which depends on the tetrahedralization of the model but remains consistently with the same insertion priority over time. We compared temporal seeding to adding seeds in index order, and adding seeds at random elements. We notice that the random seeding performs best. This is likely due to cases where temporal seeding stubbornly retains poor seeds from previous time steps that are only good for the current time step. As such, random seeding is a reasonable choice.

© 2026 Eurographics - The European Association for Computer Graphics and John Wiley & Sons Ltd.

14678659, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1111/cgf.70382 by Ecole De Technologie Superieur, Wiley Online Library on [12/06/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

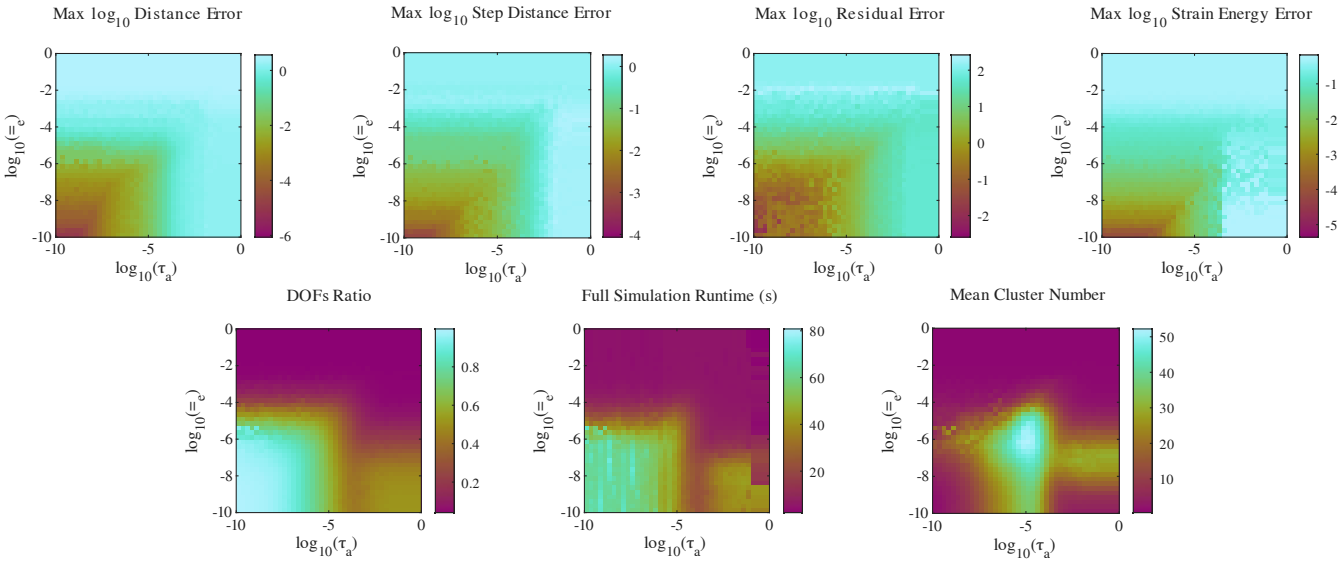


**Figure 6:** *The beam scene is simulated 1681 times in a 41 by 41 grid for a parameter sweep over thresholds to compare error and performance. Max* $\log_{10}$ *distance error: maximum vertex displacement of the affinified simulation relative to the elastic baseline across simulation frames. Max* $\log_{10}$ *step distance error : same measure, but elastic steps are reinitialized from the affinified state at each frame to discount phase drift. Max* $\log_{10}$ *residual error : Norm of full space* $Ax - b$*, representing the portion of the system that cannot be resolved with the current clusters. Max* $\log_{10}$ *strain energy : normalized difference in strain energy per step between the affinification simulation and the elastic simulation. DOFs ratio : total DOFs accumulated over all the time steps divided by the elastic baseline, indicating the overall proportion of affinification. Mean cluster number : average number of affine clusters per frame throughout the simulation.*

In Figure 6, we evaluate different accuracy and speed measures for the cantilever beam model from Figure 17 for large parameter sweep over affinification $\tau_A$ and elastification $\tau_E$ thresholds. We note that affinification thresholds below elastification thresholds lead to temporal flickering patterns alternating between affine and elastic at every time step. This adds noise below the diagonal where $\tau_e < \tau_a$. The grids visually demonstrate the accuracy-performance trade-off. Our method is subject to an inverse correlation between speed and accuracy. Increasing thresholds improves the performance but reduces the precision of the simulation through more aggressive coarsening. The simulation speed and error smoothly increase as thresholds increase, which supports the method's flexibility for different application needs. Performance is first correlated with DOFs ratios. However, there is also minor dip in performance where the mean cluster number is high suggesting a small influence of the number of clusters on the computation time. This suggests that larger affine bodies are more beneficial, while the impact of this is relatively low. This is likely because we remove affine bodies with less than two elastic elements. We typically set thresholds close to the upper diagonal of this plot where $\tau_e > tau_a$ and within an order of magnitude from each other. This matches the sweet spot near the diagonal where simulations are faster, but error is lower.

## 4.2. Performance

Table 1 shows the speedup factors achieved by affinification across different model resolutions. As expected, higher-resolution models benefit more from our method, with speedups increasing alongside the number of elements compared to the elastic baseline. This is due to the small linear cost of the oracle and clustering methods, along with cheap assembly of the reduced space mapping. We expect this phenomenon to continue at higher resolutions, and reproduced a second instance of this in the contact intensive pills scenes of the table.

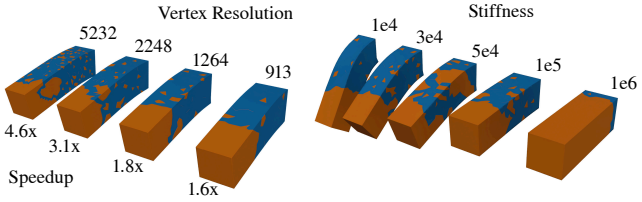


**Figure 7:** *On the left, we show a parameter sweep across mesh resolution. On the right, material stiffness.*

We also evaluate scaling on the cantilever beam example. From the results of the simulation at different mesh resolutions in Figure 7, we show how that higher resolution meshes keep the same global patterns as low resolution models, but have more opportunities for minor affinification regions, which behave linearly. In Table 2, we evaluate the same scene over higher resolution models up to more than 600k elements and 100k vertices. This test purposefully underestimates the potential benefits of our method to show scaling

**Figure 8:** *Stacked Beams affinify.*

© 2026 Eurographics - The European Association
for Computer Graphics and John Wiley & Sons Ltd.

14678659, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1111/cgf.70382 by Ecole De Technologie Superieur, Wiley Online Library on [12/06/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

**Table 1:** *We list the wall clock time aggregated over the simulation, including the overheads of the oracles and reduction. The number of elements and vertices for the pill scenes includes all 8 pills. For convenience, we also provide the speedup factors. We use either Stable Neo-Hookean (Neo) [SGK18], corotational (Corot), or Saint-Venant Kirchoff (StVK) energy models.*

| Scene | Affinification Simulation(s) | Rigidification Simulation(s) | Fully Elastic Simulation(s) | Frames | h | Speedup Affinification | Speedup Rigidification | Elements | Vertices | ν | E | Ψ | $\tau_e$ | $\tau_a$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Pills coarse | 80.37 | 128.30 | 206.18 | 4000 | 0.01 | 2.57 | 1.61 | 8760 | 2896 | 0.34<br>0.36 | 1e3<br>5e4 | Neo | 1e-4 | 1e-5 |
| Pills medium | 113.36 | 211.18 | 391.65 | 4000 | 0.01 | 3.45 | 1.85 | 14024 | 3800 | 0.34<br>0.36 | 1e3<br>5e4 | Neo | 1e-4 | 1e-5 |
| Pills HD | 187.33 | 577.17 | 1256.00 | 4000 | 0.01 | 6.71 | 2.18 | 26688 | 7872 | 0.35<br>0.36 | 1.1e3<br>5e4 | Neo | 1e-4 | 1e-5 |
| Pills UHD | 277.01 | 1089.70 | 1844.60 | 4000 | 0.01 | 10.49 | 5.23 | 41800 | 9960 | 0.35<br>0.36 | 1.1e3<br>5e4 | Neo | 1e-4 | 1e-5 |
| Octopus | 455.75 | 560.06 | 1157.70 | 2500 | 0.01 | 2.54 | 2.10 | 34820 | 10000 | 0.34 | 4e3 | Neo | 7e-5 | 3e-5 |
| Explorer | 99.39 | 133.36 | 298.87 | 2500 | 0.005 | 3.01 | 2.24 | 13204 | 3810 | 0.4<br>0.35 | 2e4<br>5e2 | Neo | 1e-3 | 1e-4 |
| Wheel | 19.65 | 49.99 | 332.44 | 1000 | 0.005 | 16.92 | 6.65 | 17450 | 5868 | 0.34<br>0.4 | 1.4e3<br>5e5 | Neo | 1e-4 | 1e-5 |
| Beam | 8.88 | 11.42 | 16.72 | 370 | 0.01 | 1.88 | 1.46 | 5516 | 1264 | 0.35 | 1e5 | StVK | 2e-4 | 1e-4 |
| | 8.87 | 11.51 | 16.45 | 370 | 0.01 | 1.85 | 1.42 | 5516 | 1264 | 0.35 | 1e5 | Neo | 2e-4 | 1e-4 |
| | 3.47 | 9.12 | 15.90 | 370 | 0.01 | 4.58 | 1.74 | 5516 | 1264 | 0.35 | 1e5 | Corot | 2e-4 | 1e-4 |
| Toothbrush | 11.95 | 16.16 | 46.01 | 600 | 0.01 | 3.85 | 2.85 | 11574 | 4325 | 0.38<br>0.38 | 5e5<br>2e2 | Neo | 1e-4 | 95e-6 |
| Armadillo Res. Limited | 38.96 | 34.66 | 111.57 | 1400 | 0.005 | 2.86 | 3.21 | 10785 | 3193 | 0.35 | 1.2e2 | Neo | 6e-12 | 4e-13 |
| Twisting beam | 2463.60 | 3224.60 | 4764.60 | 1350 | 0.01 | 1.93 | 1.48 | 642400 | 16106 | 0.38 | 5e3 | Neo | 11e-4 | 10e-4 |

**Table 2:** *Speedup factors increase with model resolution for the same affinification threshold compared to the elastic solution in the cantilever beam scene.*

| # Vertices | 913 | 1264 | 2248 | 5232 | 16106 | 44608 | 116573 |
|---|---|---|---|---|---|---|---|
| # Elements | 3803 | 5516 | 10416 | 25843 | 83639 | 244548 | 642400 |
| Speedup factor | 1.64 | 1.84 | 3.23 | 4.64 | 6.25 | 9.08 | 13.81 |

under poor parameter tuning. The same thresholds and material properties are used across resolutions leading to increasingly softer beams due to numerical precision. The speedups increase alongside resolutions, unlocking potentially boundless speedups limited only by the model resolution, even with minimal threshold tuning. This phenomenon is explained by the higher computational complexity of the Newton solve. Reducing elements at a higher scale delays combinatorial explosion, at the low cost of a linear oracle algorithm (the BFS and single PCG iteration) for elastification. On the right, stiffness has an impact on affinification as non-linearity get introduced in rotating beams. Highly stiff models have low deformation rate and so they naturally become affine quickly. Large section undergoing similar motions like the tip of each beam remain mostly coarsened throughout the tests. In Figure 9, we apply successive rotations to one end of a beam, producing a progressive twist over time. These non-linear rotations are particularly ill-suited for affinification as they cannot be fully represented as affine transformations. As the twist increases, the affinification beam becomes more and more elastic, reducing performance. This is partly due to the preconditioner for the oracle being computed at rest.

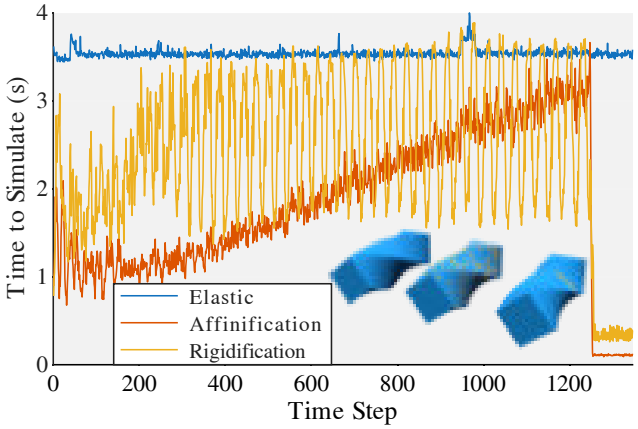


**Figure 9:** *The preconditioner for the oracle can over-elastify regions when the deformed model heavily deviates from the original pose.*

In Figure 10, we show a beam remaining affine under both compression and shearing. In Figure 1, the toothbrush is under similar motions, and so the bristles remain mostly affine. We note that because the bristles not only shear, but also bend, they require some level of elastification to represent the full range of motions. Beams are also stable when stacked on top of each other like in Figure 8, which shows large chunks of affine motions in the centre of contacting blocks.

© 2026 Eurographics - The European Association for Computer Graphics and John Wiley & Sons Ltd.

14678659, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1111/cgf.70382 by Ecole De Technologie Superieur, Wiley Online Library on [12/06/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

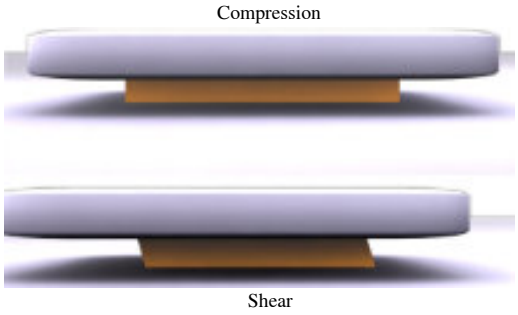


**Figure 10:** *The soft bodies remain affine under both interactions. The block remains ready for local elastification.*

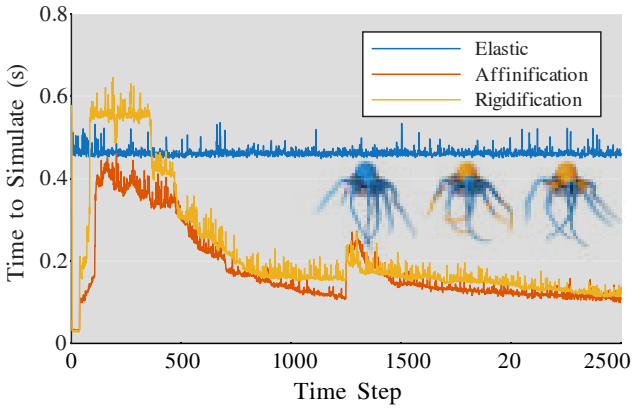


**Figure 11:** *Affinification quickly outperform the elastic simulation after the first impact. Both methods adequately wake up on interaction mid-simulation.*

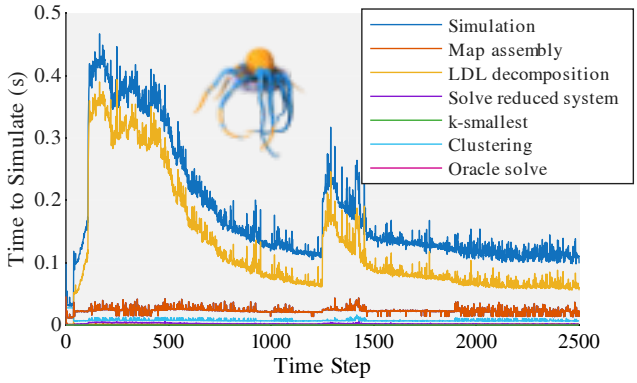


**Figure 12:** *Breakdown of the processes in the affinification simulation of the octopus scene.*

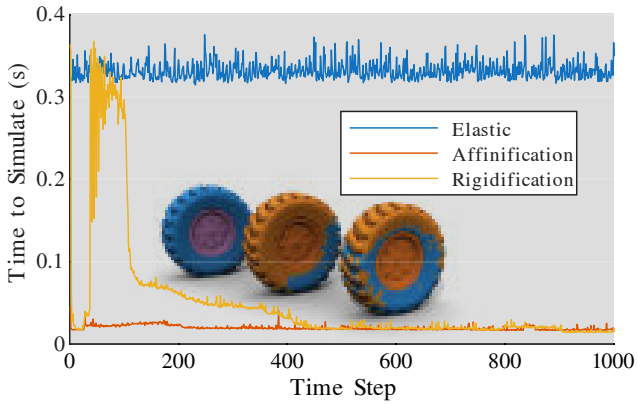


**Figure 13:** *Affinification can remain partially coarsened on impact with less distracting visual artifacts.*

### 4.3. Comparison

Adaptive rigidification is the most comparable to our approach in terms of goals, usage, and implementation. Accordingly, we conduct a detailed comparison using benchmarks examples in [MKWL22] supplemented with new scenes. In performance plots that follow, when three images are overlaid, they correspond to visualizations of the fully elastic, affinification, and rigidification simulations, respectively.

In Figure 11, an octopus is dropped in a bowl and one of its tentacles is plucked mid-simulation. Because the octopus is softer than the original version, adaptive rigidification struggles to rigidify the tentacles, but our method is able to produce visually coherent results at low cost. Our method adequately elastifies on difficult nonlinear motion like when we tug on the tentacle. Notice the initial spike for rigidification. This is caused by the first few frames being simulated as fully elastic to gather strain rate history needed for the method to work compared to affinification. The overhead of rigidification along with the nearly fully elastic octopus on impact yields simulation times higher than the elastic simulation. Our method adequatly preserves affinely deforming regions on impact, improving performance for the simulation. In Figure 12, we breakdown the different processes of affinification, including the clustering, oracle, map assembly, and more. The LDL decomposition for the time integration occupies most of the simulation time, which is expected and is the bottleneck of all the methods in our comparisons. Processes linked specifically to affinification, like the map assembly, oracle, and clustering remain a small portion of the computation and stay mostly constant across time. The key to performance lies in reducing the time for LDL decompositions by generating a smaller system.

In Figure 13, we showcase the wheel scene where affinification is able to retain coarsened regions near the top on impact while rigidification wakes completely for a few frames. Once settled, the performances remain similar. Near the end, the performances overlap up to a few degrees of freedom in difference. This example also showcases that heterogeneous materials are compatible with the method. In theory, mapping elastic elements of different stiffness to the same affine body would stiffen it. Practically, this has little impact on the behaviour of the model as thin layers of elastic elements form at the boundaries between materials when they diverge too significantly.

© 2026 Eurographics - The European Association
for Computer Graphics and John Wiley & Sons Ltd.

14678659, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1111/cgf.70382 by Ecole De Technologie Superieur, Wiley Online Library on [12/06/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

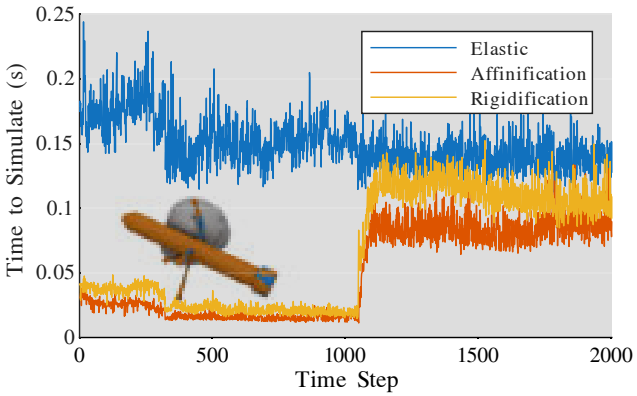


**Figure 14:** *The explorer 1 satellite is struck by a jello meteor. Everything but the antennas move rigidly before impact.*

In Figure 14 we present the explorer 1 satellite undergoing mostly rotational motions. The antennas flop around elastically, which gets partially affine. However, we note that, unlike rigidification, affinification is subject to numerical damping like the elastic simulation. As such, the model is not able to keep spinning forever. The performances of rigidification and affinification remain similar in this example, but a slight edge favours affinification due to the affine parts of the antennas.

We also tested scaling on a contact-heavy scene. The Figure 15 shows three versions of the same model with different tetrahedralizations from coarse to fine, respectively. We notice that increasing the resolution flattens the affinification plot line compared to rigidification. We suspect that this is because motions tend to become linear as the elements become infinitely small. As such, there are more opportunities for affinification at a finer scale.

For fairness, we disabled resource capping for all of our tests except for the armadillo scene of Figure 16. In that scene, we compare the elastic simulation to affinification and rigidification with a minimum of 33% coarsening, or about a third of the elements that must remain affine. In the unlikely scenario that every element deforms at the same time, like in this scene, the reduced DOFs will still provide decent linear behaviour for a step through the affine transform, which with rigidification instead yields major visually distracting artifacts.

All the videos in the supplemental materials show that affinification better preserves fine-scale deformations than rigidification, particularly in regions undergoing shearing or non-uniform stretching. Notably, the toothbrush and sandwich scenes highlight the method's ability to capture subtle material behaviours that rigidification cannot represent.

In Figure 17, we compare the elastic simulation, which we consider to be the ground truth, to elastification and rigidification. With the right thresholds, both methods can be visually indistinguishable from the elastic simulation. In both cases, the performance can be significantly improved while preserving the visual coherence. Because we did not implement pinned constraints on affine bodies,

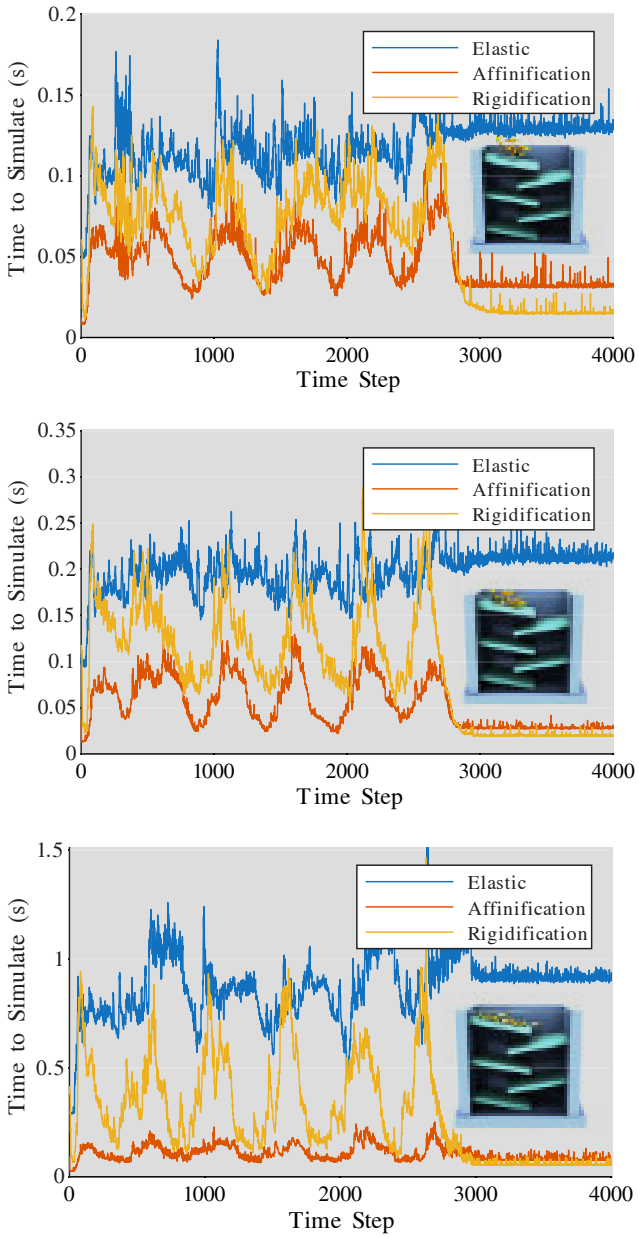


**Figure 15:** *Pills with resolutions from coarse to fine. Finer models offer more opportunities for affinification, flattening the plot as resolutions increase.*

we let the pinned ends of the cantilever remain elastic so it stays in place.

## 5. Discussion and Limitations

We use a BFS which remains consistent with adaptive rigidification, yet there are opportunities to use parallelization algorithms for clustering [GPP*14; CNS*18]. For instance, it is possible to do a bottom-up approach by merging clusters under shared

© 2026 Eurographics - The European Association for Computer Graphics and John Wiley & Sons Ltd.

14678659, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1111/cgf.70382 by Ecole De Technologie Superieur, Wiley Online Library on [12/06/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

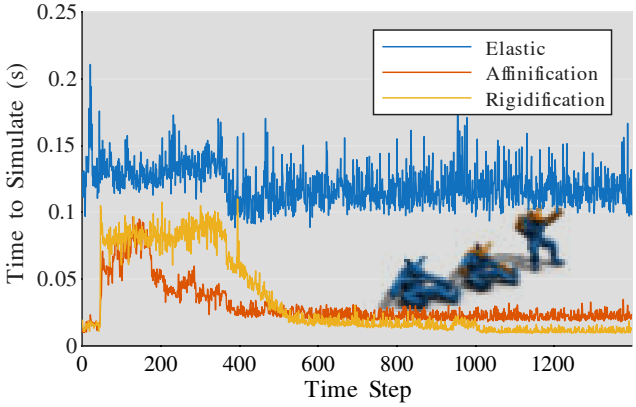


**Figure 16:** *The time to simulate is restricted by the minimum number of coarsened elements.*

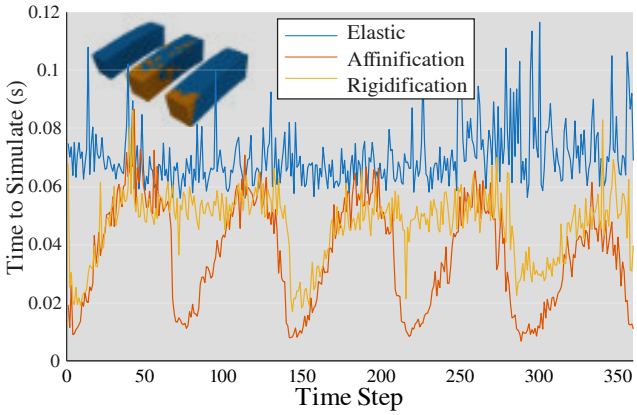


**Figure 17:** *Performances oscillate as parts come in and out of affine space.*

affine transforms. However, such an approach, while potentially extremely efficient, would not change the complexity of the prediction algorithm.

While our method works with heterogeneous materials as shown in multiple examples, we note that during phases where stiff and soft materials share an affine body, the stiffness gets dominated by the affine body. Such phenomena happen due to our way of assembling the stiffness of affine bodies through a projection of the full space stiffness. This naturally leads to elastic boundaries often appearing near the affine stiff regions. It would potentially be possible to instead assemble the stiffness of each affine body directly and reduce the impact of combining different materials.

While manual assignment of ABD and FEM regions could be manageable for some scenarios presented like the toothbrush, our method automates this process, reducing manual effort and enabling faster iteration. Moreover, the partially affine regions benefit from speed ups without compromising realism unlike predefined regions that cannot be as expressive. Our simulation is ready for full elastification at any point should the need arise through non-affine deformation like large contacts, as shown in examples like the bowling balls falling on a mattress.

In the current implementation, we do not terminate the clustering if min-max bounds drift. It may be reasonable to terminate the clusters when $\|F_{min} - F_{max}\| > 2\tau$ to reduce potential divergence of clusters from the thresholds.

While the precomputed preconditioner is efficient, it may provide a poor approximation of the inverse matrix when the system is overly deformed like in large twisting motions. This has the effect of increasing elastification, reducing performance. Future work could explore adaptive threshold selection, increasing thresholds as the quality of the preconditioner decreases.

Although optimal partitioning strategies could increase the cluster sizes, computing globally optimal clusters would require solving a non-trivial optimisation problem at each time step, which introduces considerable overhead. This additional cost could outweigh the benefits, as the expected reduction in error is likely marginal compared to the performance gains already achieved with our deformation-rate-based greedy clustering. For high-resolution models, maintaining linear complexity is critical to scalability.

We opted not to use a machine learning algorithm for this work as it would not lead to an improvement in complexity for the oracle's prediction of deformation. This would still require a linear pass over elements to predict deformation, though it might be interesting to explore novel learned oracles in the future.

## 6. Conclusion

We have presented affinification, a novel method for accelerating physics-based animation. By allowing coarsened regions to undergo scaling, shearing, and non-uniform deformation, our approach bridges the gap between rigid and elastic representations, enabling more expressive and accurate simulations at significantly reduced computational cost. Through extensive evaluation across diverse benchmark scenes, we demonstrate that affinification consistently outperforms both fully elastic simulations and adaptive rigidification in terms of speed, while preserving or improving visual fidelity. Our method achieves up to 16× speedups, scales favourably with model resolution, and remains robust under challenging conditions.

Ablation studies further validate the effectiveness of our clustering and prediction strategies. Our optional resource capping method enhances the applicability of our method to resource-constrained applications through the predictability of the time to integrate motions.

Affinification offers a flexible and scalable framework for model reduction in deformable simulation. It opens the door to more aggressive coarsening strategies without overly compromising realism, and lays the groundwork for future extensions in collision detection, parallel clustering, and learned deformation prediction. We believe this work represents a meaningful step toward unifying performance and fidelity in physics-based animation, and we look forward to its adoption and further development by the graphics community.

© 2026 Eurographics - The European Association
for Computer Graphics and John Wiley & Sons Ltd.

14678659, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1111/cgf.70382 by Ecole De Technologie Superieur, Wiley Online Library on [12/06/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

**Acknowledgements**

The Bellairs Workshop on Computer Animation was instrumental in the conception of the research presented in this paper. This work was partially supported by the NSERC grants DGECR-2021-00461, RG-PIN 2021-03707, RG-PIN 2024-04523, FRQNT 2023-NOVA-315274, NSERC ALLRP 571411-21, as well as a gift from Adobe Research. This work was also supported in part through the Digital Research Alliance of Canada resources, services, and staff expertise.

## References

[AR12] ARTEMOVA, SVETLANA and REDON, STEPHANE. "Adaptively Restrained Particle Simulations". *Phys. Rev. Lett.* 109 (19 2012), 190201 2.

[BC96] BRO-NIELSEN, MORTEN and COTIN, STEPHANE. "Real-time Volumetric Deformable Models for Surgery Simulation using Finite Elements and Condensation". *Computer Graphics Forum* 15.3 (1996), 57–66. DOI: 10.1111/1467-8659.1530057 2.

[BJ05] BARBIČ, JERNEJ and JAMES, DOUG L. "Real-Time Subspace Integration for St. Venant-Kirchhoff Deformable Models". *ACM Trans. Graph.* 24.3 (July 2005), 982–990. ISSN: 0730-0301. DOI: 10.1145/1073204.1073300 2.

[BW98] BARAFF, DAVID and WITKIN, ANDREW. "Large Steps in Cloth Simulation". *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH 1998. New York, NY, USA: Association for Computing Machinery, 1998, 43–54. ISBN: 0897919998. DOI: 10.1145/280814.280821 3.

[BZA14] BUDSBERG, JEFF, ZAFAR, NAFEES BIN, and ALDÉN, MIHAI. "Elastic and plastic deformations with rigid body dynamics". *ACM SIGGRAPH 2014 Talks*. SIGGRAPH '14. Vancouver, Canada: Association for Computing Machinery, 2014. ISBN: 9781450329606. DOI: 10.1145/2614106.2614132. URL: https://doi.org/10.1145/2614106.2614132 2.

[CBK20] COEVOET, EULALIE, BENCHEKROUN, OTMAN, and KRY, PAUL G. "Adaptive Merging for Rigid Body Simulation". *ACM Trans. Graph.* 39.4 (2020) 2.

[CNS*18] CHEN, JUN, NONAKA, KEISUKE, SANKOH, HIROSHI, et al. "Efficient Parallel Connected Component Labeling With a Coarse-to-Fine Strategy". *IEEE Access* 6 (2018), 55731–55740. DOI: 10.1109/ACCESS.2018.2872452 10.

[DDCB01] DEBUNNE, GILLES, DESBRUN, MATHIEU, CANI, MARIE-PAULE, and BARR, ALAN H. "Dynamic Real-time Deformations Using Space & Amp; Time Adaptive Sampling". *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH '01. New York, NY, USA: ACM, 2001, 31–36. ISBN: 1-58113-374-X. DOI: 10.1145/383259.383262 2.

[Erl04] ERLEBEN, KENNY. "Stable, robust, and versatile multibody dynamics animation". PhD thesis. University of Copenhagen, 2004 3.

[FMD*19] FULTON, LAWSON, MODI, VISMAY, DUVENAUD, DAVID, et al. "Latent-space Dynamics for Reduced Deformable Simulation". *Computer Graphics Forum* 38.2 (2019), 379–391. DOI: https://doi.org/10.1111/cgf.13645. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.13645. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.13645 2.

[FSKP23] FERGUSON, ZACHARY, SCHNEIDER, TESEO, KAUFMAN, DANNY, and PANOZZO, DANIELE. "In-Timestep Remeshing for Contacting Elastodynamics". *ACM Trans. Graph.* 42.4 (July 2023). ISSN: 0730-0301. DOI: 10.1145/3592428. URL: https://doi.org/10.1145/3592428 2.

[GKS02] GRINSPUN, EITAN, KRYSL, PETR, and SCHRÖDER, PETER. "CHARMS: A Simple Framework for Adaptive Simulation". *ACM Trans. Graph.* 21.3 (2002), 281–290. ISSN: 0730-0301. DOI: 10.1145/566654.566578 2.

[GPP*14] GUPTA, SIDDHARTH, PALSETIA, DIANA, PATWARY, MD. MOSTOFA ALI, et al. "A New Parallel Algorithm for Two-Pass Connected Component Labeling". *2014 IEEE International Parallel & Distributed Processing Symposium Workshops*. Phoenix, AZ, USA: IEEE, 2014, 1355–1362. DOI: 10.1109/IPDPSW.2014.152 10.

[HTC*14] HAHN, FABIAN, THOMASZEWSKI, BERNHARD, COROS, STELIAN, et al. "Subspace clothing simulation using adaptive bases". *ACM Trans. Graph.* 33.4 (July 2014). ISSN: 0730-0301. DOI: 10.1145/2601097.2601160. URL: https://doi.org/10.1145/2601097.2601160 2.

[KKZZ19] KAPLAN, HAIM, KOZMA, LÁSZLÓ, ZAMIR, OR, and ZWICK, URI. "Selection from Heaps, Row-Sorted Matrices, and X+Y Using Soft Heaps". *2nd Symposium on Simplicity in Algorithms (SOSA 2019)*. Ed. by FINEMAN, JEREMY T. and MITZENMACHER, MICHAEL. Vol. 69. Open Access Series in Informatics (OASIcs). Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2019, 5:1–5:21. ISBN: 978-3-95977-099-6. DOI: 10.4230/OASIcs.SOSA.2019.5. URL: https://drops.dagstuhl.de/entities/document/10.4230/OASIcs.SOSA.2019.5 6.

[LKL*22] LAN, LEI, KAUFMAN, DANNY M., LI, MINCHEN, et al. "Affine body dynamics: fast, stable and intersection-free simulation of stiff materials". *ACM Trans. Graph.* 41.4 (July 2022). ISSN: 0730-0301. DOI: 10.1145/3528223.3530064 3, 5.

[LLL*25] LU, ZIXUAN, LIU, ZIHENG, LAN, LEI, et al. "High-performance CPU Cloth Simulation Using Domain-decomposed Projective Dynamics". *ACM Trans. Graph.* 44.4 (July 2025). ISSN: 0730-0301. DOI: 10.1145/3731182. URL: https://doi.org/10.1145/3731182 1.

[Mer25] MERCIER-AUBIN, ALEXANDRE. "Adaptive Methods for Elastic Deformation". PhD thesis. McGill University, 2025 3.

[MFRC13] MANTEAUX, PIERRE-LUC, FAURE, FRANÇOIS, REDON, STEPHANE, and CANI, MARIE-PAULE. "Exploring the Use of Adaptively Restrained Particles for Graphics Simulations". *VRIPHYS 2013 - 10th Workshop on Virtual Reality Interaction and Physical Simulation*. Lille, France: Eurographics Association, Nov. 2013, 17–24. DOI: 10.2312/PE.vriphys.vriphys13.017-024 2.

[MK23] MERCIER-AUBIN, ALEXANDRE and KRY, PAUL G. "Adaptive Rigidification of Discrete Shells". *Proc. ACM Comput. Graph. Interact. Tech.* 6.3 (Aug. 2023). DOI: 10.1145/3606932 2, 5.

[MK24] MERCIER-AUBIN, ALEXANDRE and KRY, PAUL G. "A Multilayer Solver for XPBD". *Computer Graphics Forum* 43.8 (2024). ISSN: 1467-8659. DOI: 10.1111/cgf.15186 3, 5.

[MKWL22] MERCIER-AUBIN, ALEXANDRE, KRY, PAUL G., WINTER, ALEXANDRE, and LEVIN, DAVID I. W. "Adaptive Rigidification of Elastic Solids". *ACM Trans. Graph.* 41.4 (July 2022). ISSN: 0730-0301. DOI: 10.1145/3528223.3530124 2, 5, 9.

[Mül08] MÜLLER, MATTHIAS. "Hierarchical Position Based Dynamics". *Workshop on Virtual Reality Interaction and Physical Simulation VRIPHYS 2008*. Vol. 8. Grenoble, France: The Eurographics Association, Jan. 2008, 1–10. DOI: 10.2312/PE/vriphys/vriphys08/001-010 3.

[NKJF09] NESME, MATTHIEU, KRY, PAUL G., JEŘÁBKOVÁ, LENKA, and FAURE, FRANÇOIS. "Preserving Topology and Elasticity for Embedded Deformable Models". *ACM Trans. Graph.* 28.3 (July 2009). ISSN: 0730-0301. DOI: 10.1145/1531326.1531358 2.

[NSO12] NARAIN, RAHUL, SAMII, ARMIN, and O'BRIEN, JAMES F. "Adaptive Anisotropic Remeshing for Cloth Simulation". *ACM Trans. Graph.* 31.6 (2012), 152:1–152:10. ISSN: 0730-0301 2.

[NSPO14] NARAIN, RAHUL, SAMII, ARMIN, PFAFF, TOBIAS, and O'BRIEN, J. "ARCSim: Adaptive refining and coarsening simulator". *University of California—Berkeley, accessed Oct* 1 (2014), 2016 2.

© 2026 Eurographics - The European Association for Computer Graphics and John Wiley & Sons Ltd.

14678659, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1111/cgf.70382 by Ecole De Technologie Superieur, Wiley Online Library on [12/06/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

[SGK18] SMITH, BREANNAN, GOES, FERNANDO DE, and KIM, THEODORE. "Stable Neo-Hookean Flesh Simulation". *ACM Trans. Graph.* 37.2 (Mar. 2018). ISSN: 0730-0301. DOI: 10.1145/3180491. URL: https://doi.org/10.1145/3180491 8.

[SRH*16] SCHRECK, CAMILLE, ROHMER, DAMIEN, HAHMANN, STEFANIE, et al. "Nonsmooth Developable Geometry for Interactively Animating Paper Crumpling". *ACM Trans. Graph.* 35.1 (Dec. 2016). ISSN: 0730-0301. DOI: 10.1145/2829948 2.

[SYS*21] SHEN, SIYUAN, YANG, YIN, SHAO, TIANJIA, et al. "High-order differentiable autoencoder for nonlinear model reduction". *ACM Trans. Graph.* 40.4 (July 2021). ISSN: 0730-0301. DOI: 10.1145/3450626.3459754. URL: https://doi.org/10.1145/3450626.3459754 2.

[TFLK24] TRUSTY, TY, FEI, YUN (RAYMOND), LEVIN, DAVID, and KAUFMAN, DANNY. "Trading Spaces: Adaptive Subspace Time Integration for Contacting Elastodynamics". *ACM Trans. Graph.* 43.6 (Nov. 2024). ISSN: 0730-0301. DOI: 10.1145/3687946 2.

[TMDK15] TENG, YUN, MEYER, MARK, DEROSE, TONY, and KIM, THEODORE. "Subspace Condensation: Full Space Adaptivity for Subspace Deformations". *ACM Trans. Graph.* 34.4 (July 2015). ISSN: 0730-0301. DOI: 10.1145/2766904 2.

[TNFG14] TOURNIER, MAXIME, NESME, MATTHIEU, FAURE, FRANCOIS, and GILLES, BENJAMIN. "Seamless adaptivity of elastic models". *Proceedings of Graphics Interface 2014*. GI 2014. Montréal, Québec, Canada: Canadian Human-Computer Communications Society, 2014, 17–24. ISBN: 978-1-4822-6003-8 2.

[WZQ*17] WANG, MEILI, ZHENG, HUA, QIAN, KUN, et al. "Sampling Hierarchical Position-Based Dynamics Simulation". *Next Generation Computer Animation Techniques*. Ed. by CHANG, JIAN, ZHANG, JIAN JUN, MAGNENAT THALMANN, NADIA, et al. Cham: Springer International Publishing, 2017, 45–55. ISBN: 978-3-319-69487-0 3.

[ZDF*22] ZHANG, JIAYI ERIS, DUMAS, JÉRÉMIE, FEI, YUN, et al. "Progressive simulation for cloth quasistatics". *ACM Transactions on Graphics (TOG)* 41.6 (2022), 1–16 2.

© 2026 Eurographics - The European Association for Computer Graphics and John Wiley & Sons Ltd.

14678659, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.1111/cgf.70382 by Ecole De Technologie Superieur, Wiley Online Library on [12/06/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License