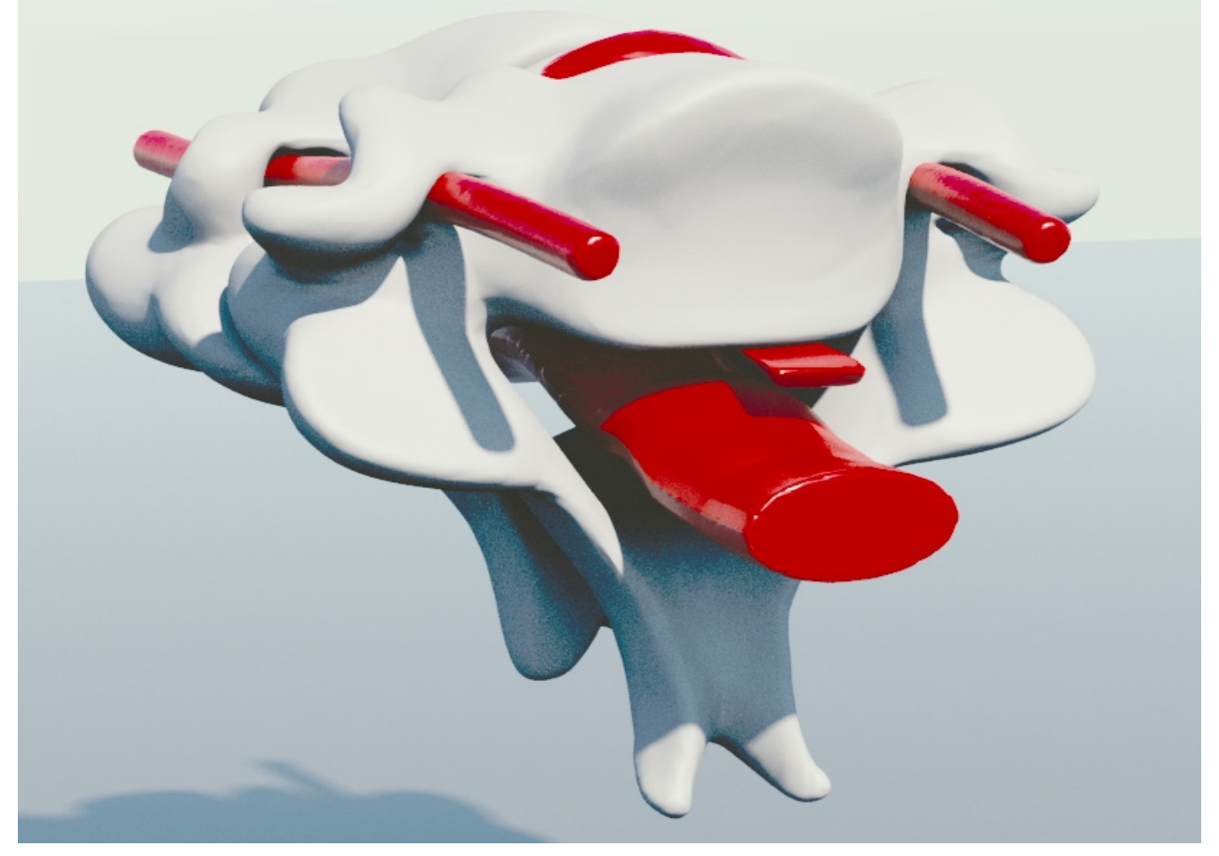
Interactive Soft-body Simulation for Surgical Applications

Nabil Boukadida, Sheldon Andrews, and Eric Paquette

École de technologie supérieure

Introduction

This is a presentation of our ongoing work for physics-based simulation of soft bodies using Extended Position Based Dynamics (XPBD). This approach is stable and efficient, and achieves interactive framerates, even on mobile devices. Strain constraints, which are inspired by the Finite Element Method (FEM), are used to simulate the physical behavior of deformable objects such as muscle tissue and human organs. Anderson acceleration is applied to the solver to boost the convergence. A graph coloring technique is also used to accelerate the convergence. This approach runs in real time on the GPU.

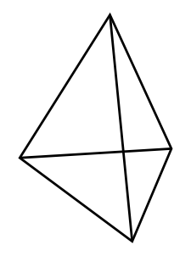


XPBD

We use XPBD [1] for our simulation. Strain constraints are applied on particle positions $\boldsymbol{x}$ which are projected in parallel in a Jacobi fashion on the GPU by solving a linear system :

$$\left[\nabla C(x) M^{-1} \nabla C(x)^T + \tilde{\alpha}\right] \Delta\lambda_i = -C(x) - \tilde{\alpha}\lambda_i \,,$$

where $\tilde{\alpha}$ describes the compliance, $\boldsymbol{M}$ is the diagonal mass matrix, and the constraint $\boldsymbol{C}$ is defined by the green strain tensor:



$$C(p_1, p_2, p_3, p_4) = \varepsilon = \frac{1}{2}\left(F^T F - I\right).$$

Numerical Acceleration

We explored some numerical acceleration methods to reduce the number of iterations required to reach convergence. Specifically, the following methods were evaluated:

1- Successive-Over-Relaxation (SOR)

This method consists of multiplying, at each iteration, the displacement vector $\Delta\boldsymbol{x}$ by a factor $\boldsymbol{\alpha} > 1$.

2 - Chebyshev acceleration

This method [2] uses Chebyshev polynomials to minimize the current error of the iterate. By simply weighting the current solution and the previous one, it allows for faster convergence.

3 - Anderson acceleration

Anderson acceleration [3] relies on the previous m solutions to estimate the next solution. It treats the sequence produced by the solver as a fixed point iteration algorithm where the goal is to minimize the residual function:

$$R\left(x^{n+1}\right) = x^{n+1} - x^{n} \ .$$

Then, by solving the following linear least-squares:

$$\omega^* = arg\min_{\omega} \left\| R^k + \sum_{j=1}^{m} \omega_j (R^{k-j} - R^k) \right\|$$

we obtain the coordinates of the new $\boldsymbol{x}$ positions within the affine subspace defined by the m previous iterates where the residual $\boldsymbol{R}$ is minimum.

Unlike SOR and Chebyshev, Anderson acceleration has remarkably improved the convergence of the solver as shown in Fig. 3 and 4. It was established that for the same number of iterations, the Anderson acceleration was able to converge more quickly. In fact, the Anderson method was able to achieve the same convergence with less than 10% of the iterations required without acceleration.

Graph Coloring

The Jacobi solver allows for a straightforward parallel implementation. However, it suffers from a low convergence rate. We use a graph coloring method [4] to boost the convergence further (Fig. 1 and 2). During the initialization phase, a graph G of constraints is constructed, where the nodes are interconnected based on sets of shared particles. Next, a detection of the set of cliques is applied and cliques are seperated. Within each group, the constraints are solved in parallel, whereas groups of colors are solved sequentially in a Gauss-Seidel fashion.

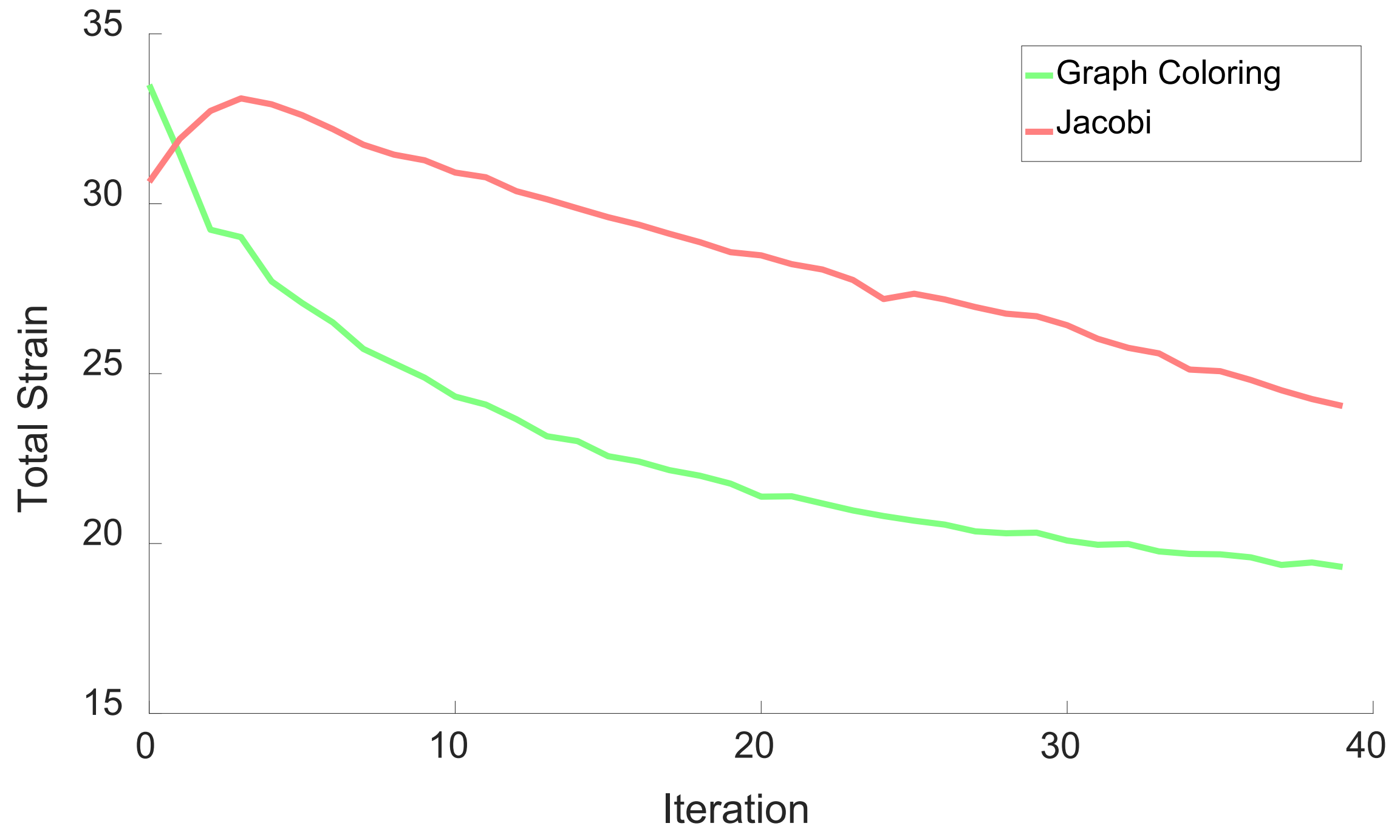


Figure 1: Convergence with and without graph coloring.

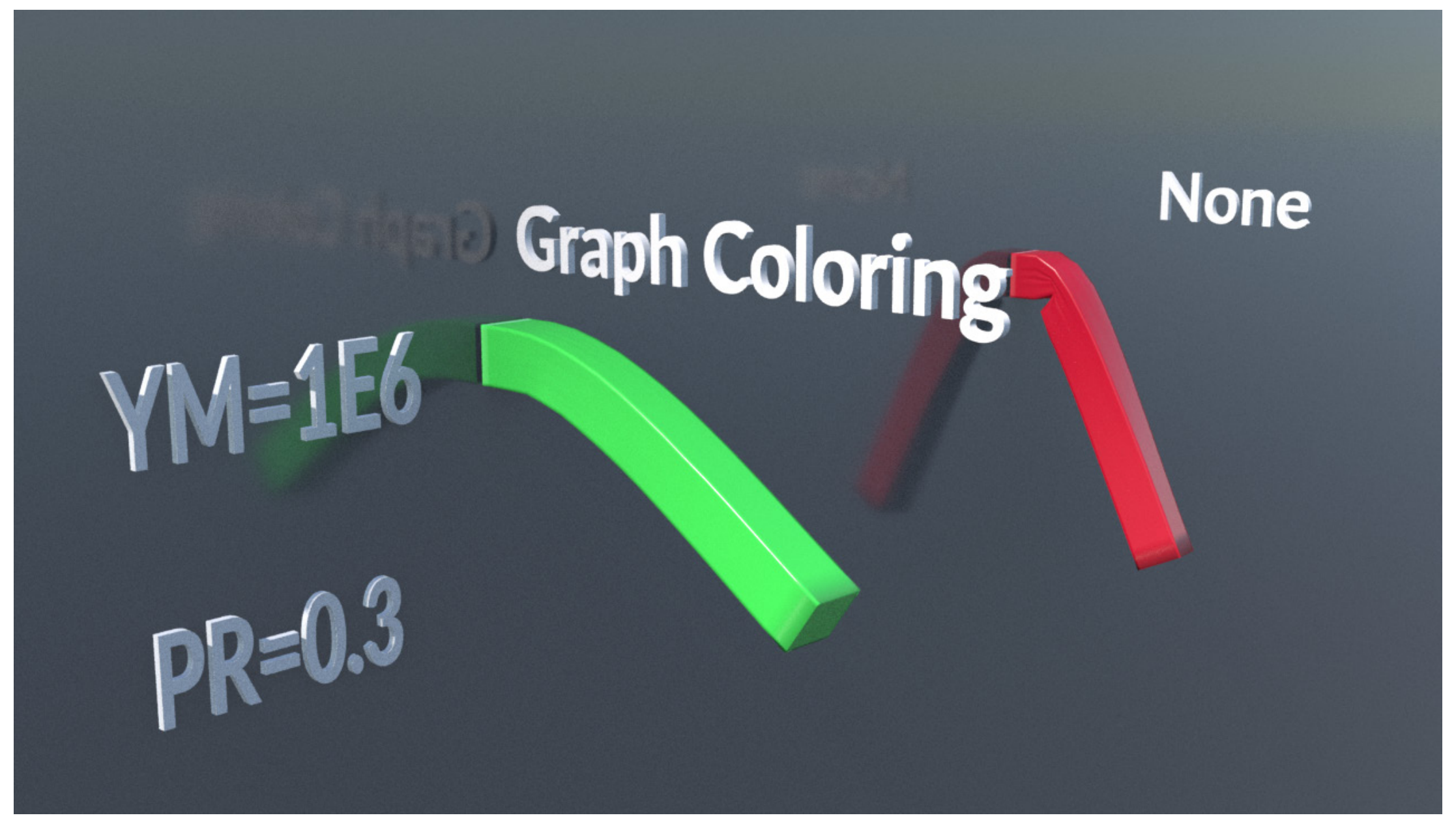


Figure 2: Simulation with and without graph coloring (40 iterations).

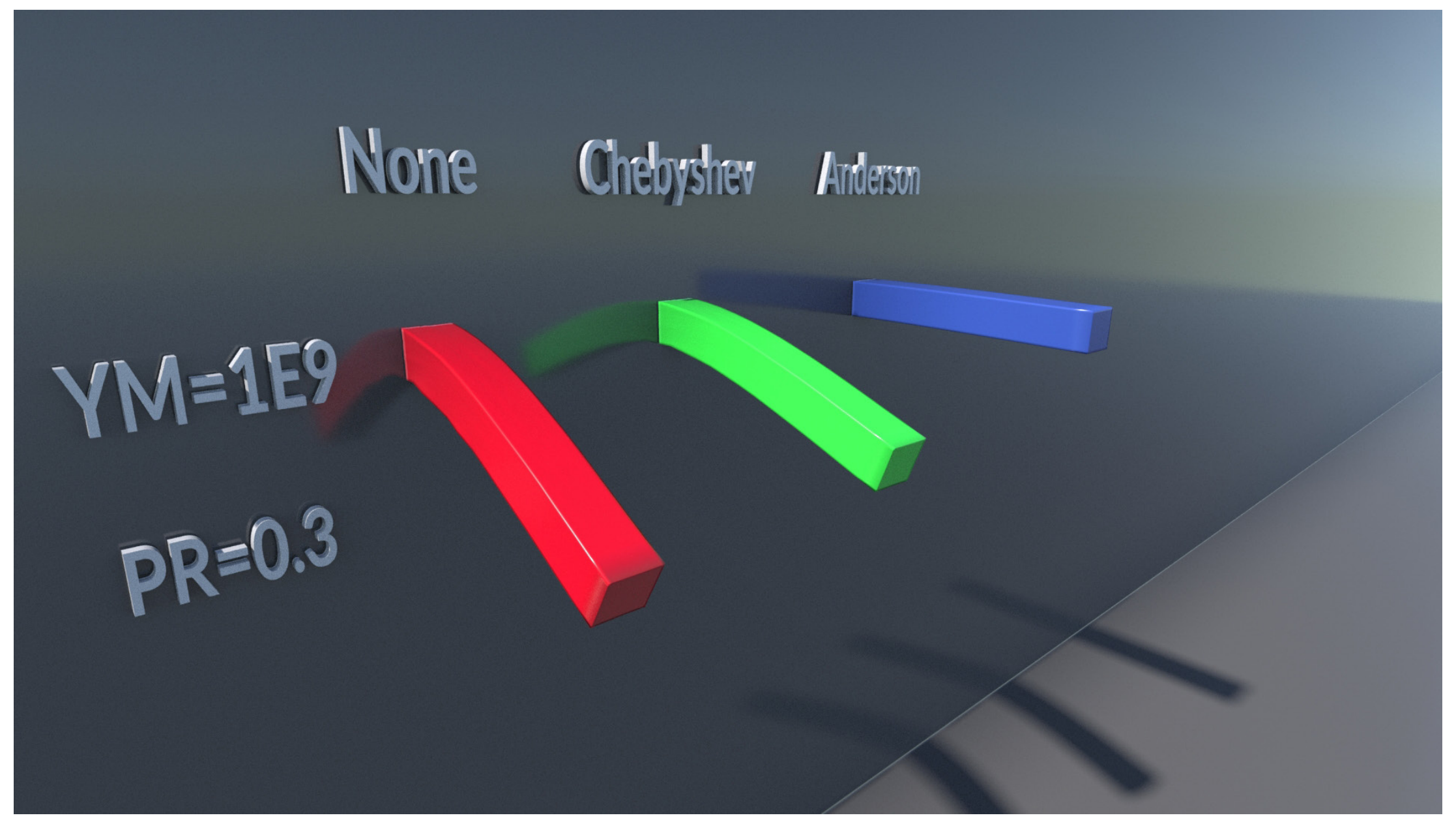


Figure 3: Simulation with and without acceleration (40 iterations).

Discussion & Future Work

We observe that although Anderson acceleration improves convergence, it artificially increases the stiffness of the material, regardless of the Young's modulus: it is stiffer as the number of iterations is increased. We plan to address this limitation by applying the acceleration on the Lagrangian multipliers instead of positions. Another concern is that, coupling the Anderson acceleration with graph coloring, has resulted in some instabilities during the simulation.

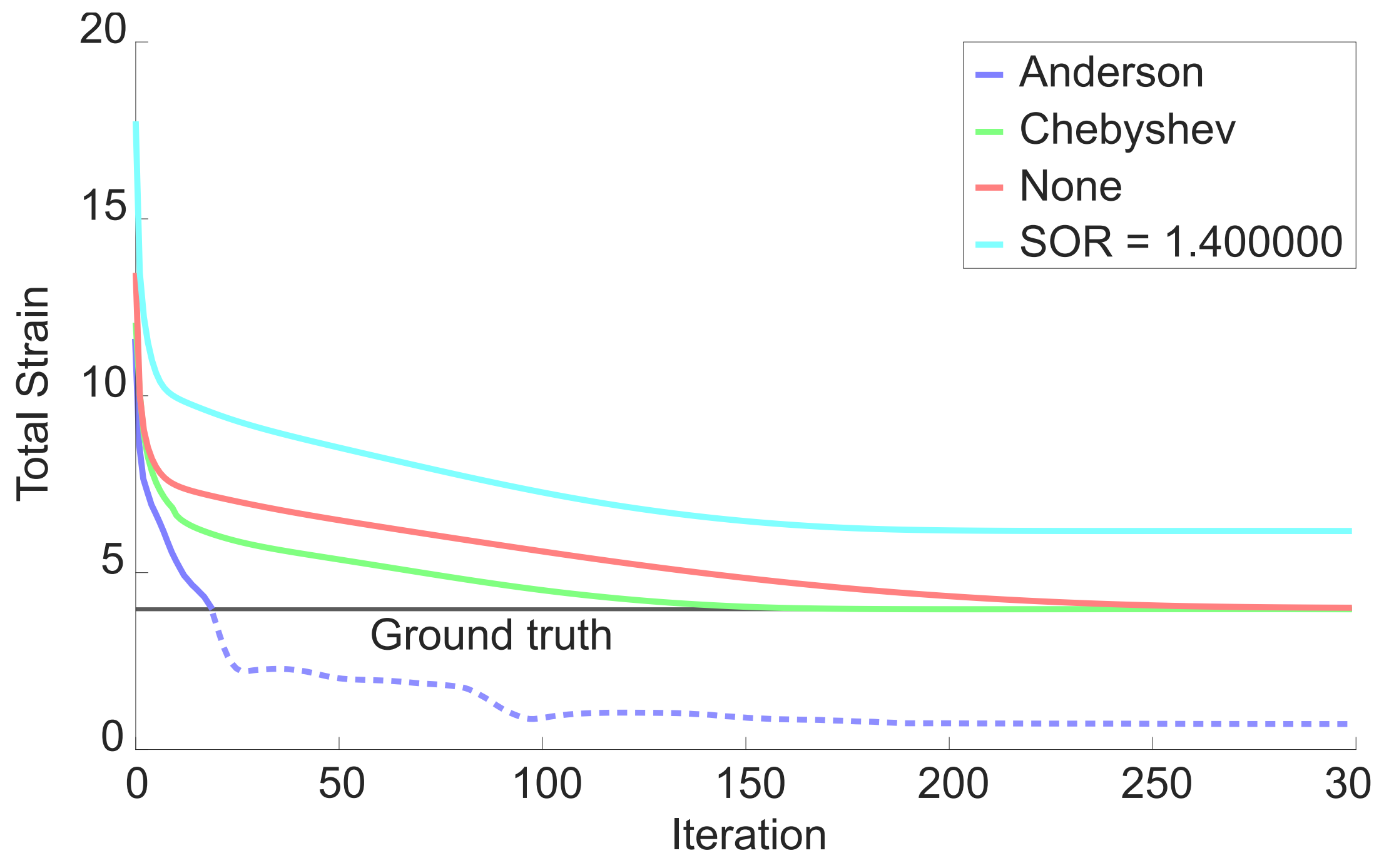


Figure 4: Convergence with numerical acceleration / Anderson converging to PBD solution.

Example videos: http://bit.do/eStKB

References

[1] Macklin, M. et al. 2016. XPBD: position-based simulation of compliant constrained dynamics. Proceedings of the 9th International Conference on Motion in Games - MIG '16, pp. 49–54.

[2] Wang, H. 2015. A chebyshev semi-iterative approach for accelerating projective and position-based dynamics. ACM Transactions on Graphics. 34(6), pp. 1–9.

[3] Peng, Y. et al. 2018. Anderson Acceleration for Geometry Optimization and Physics Simulation. ACM Transactions on Graphics. 37(4), pp. 1–14.

[4] Fratarcangeli, M. and Pellacini, F. 2015. Scalable Partitioning for Parallel Position Based Dynamics. Computer Graphics Forum. 34(2), pp. 405–413.