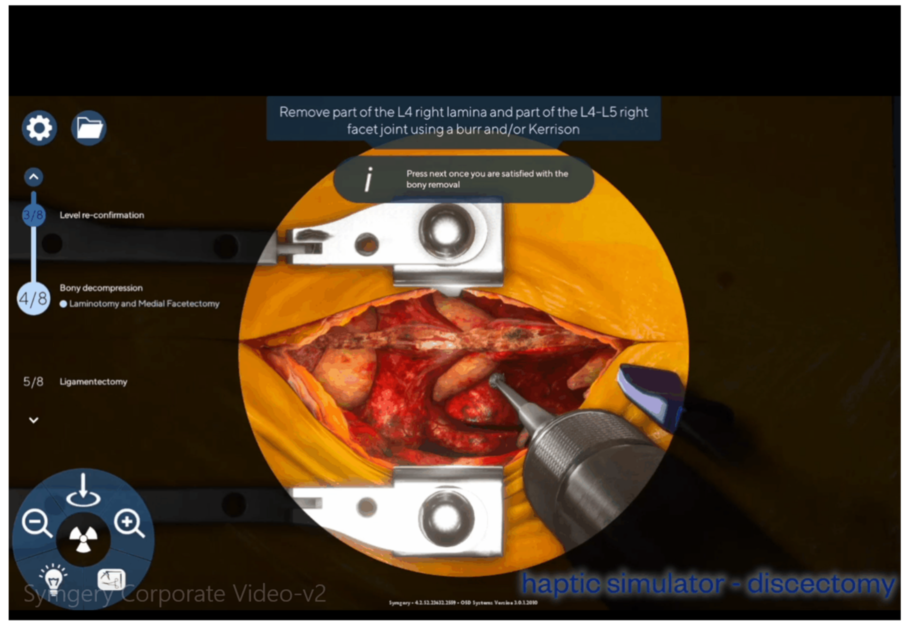
Parallel Block Neo-Hookean XPBD using Graph Clustering

Quoc-Minh Ton-That[1], Paul G. Kry[2], Sheldon Andrews[1]

[1]École de Technologie Supérieure [2]McGill University

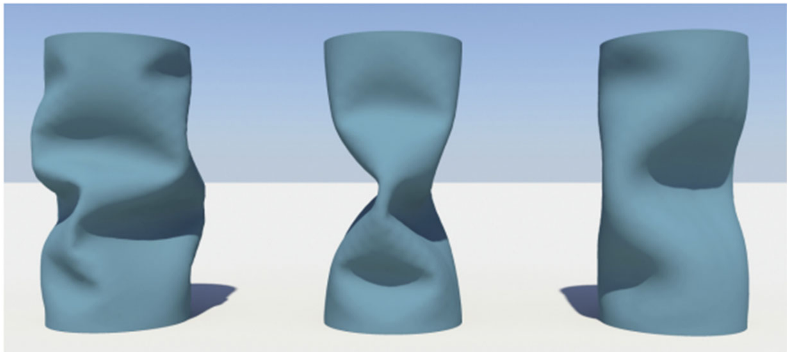
Accelerating Soft Body Simulation

- Why do we care?
  - Real-time applications
    - Virtual surgery
    - Computer games
  - Off-line applications
    - Visual effects
    - Robotics

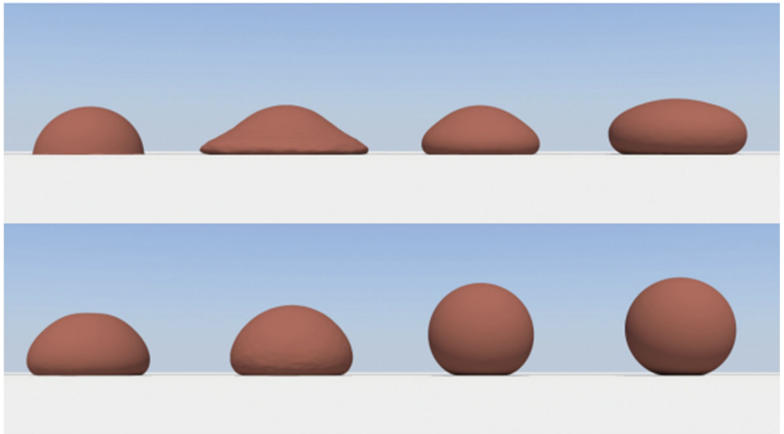


Symgery Inc., 2021

Existing Methods: Projective Dynamics

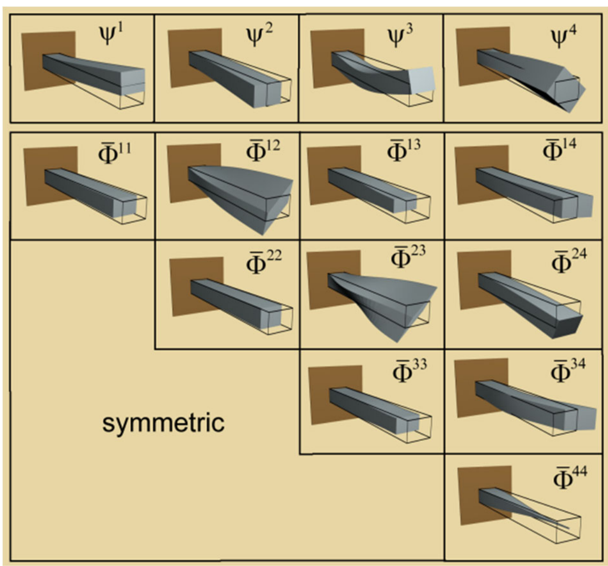


Bouaziz et al., 2014

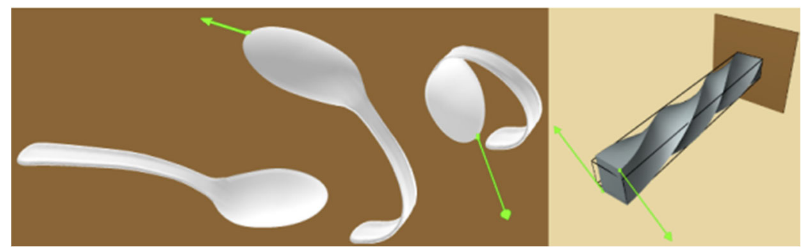


Bouaziz et al., 2014

Existing Methods: Model Reduction

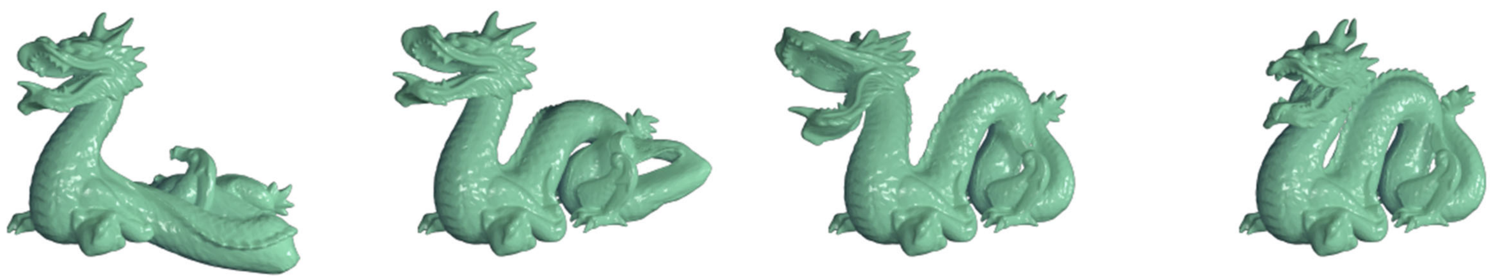


Barbic et al. 2005



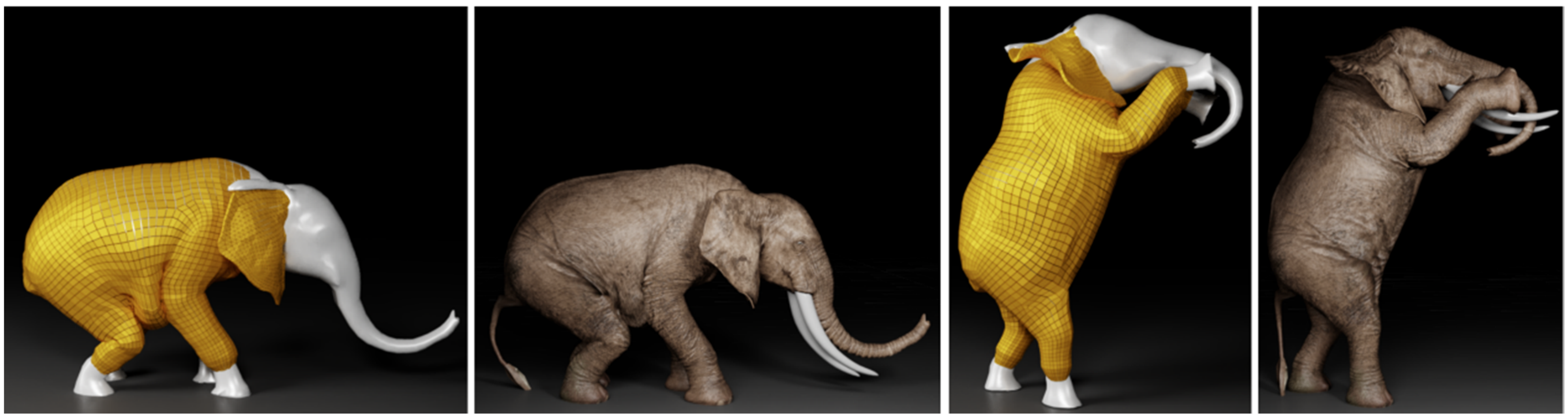
Barbic et al. 2005

Existing Methods: Multigrid Methods



Xian et al. 2019

Extended Position Based Dynamics (XPBD)

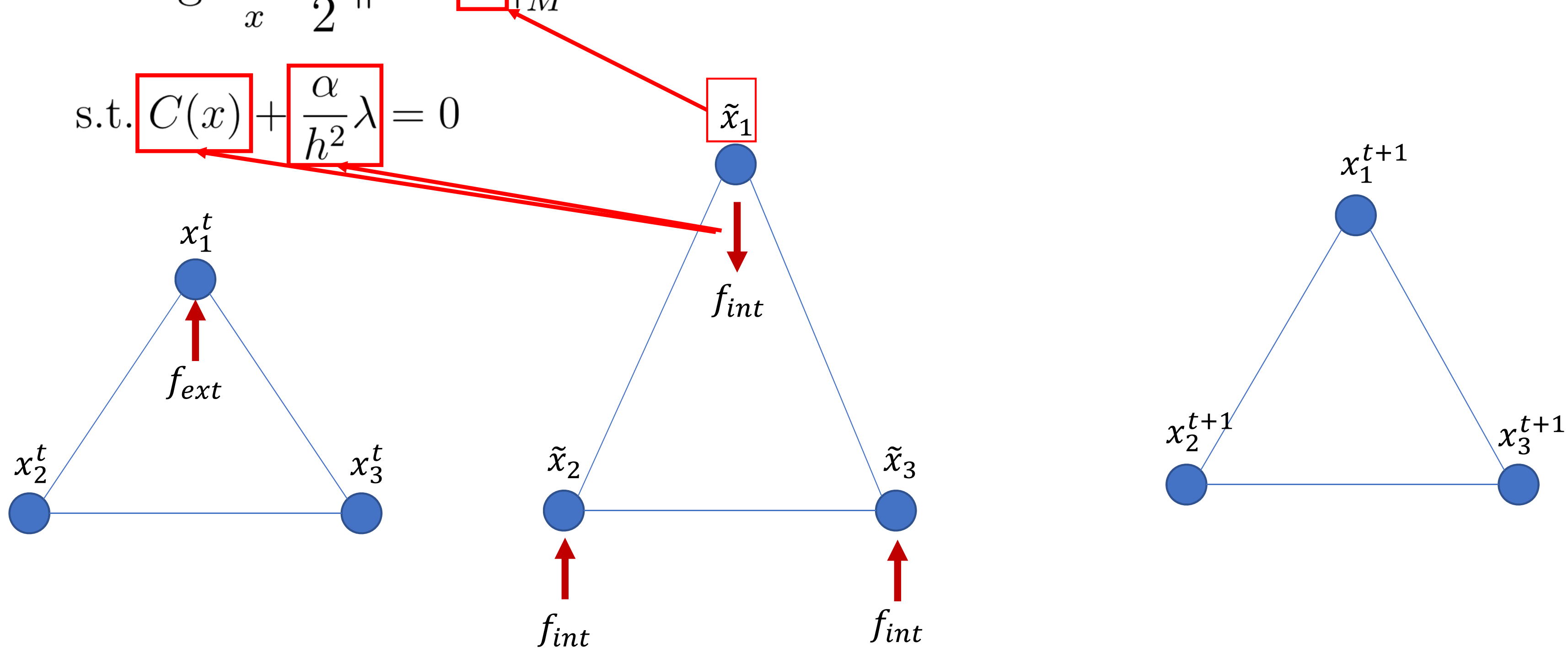


Macklin et al. 2021

XPBD in a nutshell

$$x^{t+1} = \arg\min_x \frac{1}{2} \|x - \tilde{x}\|_M^2$$

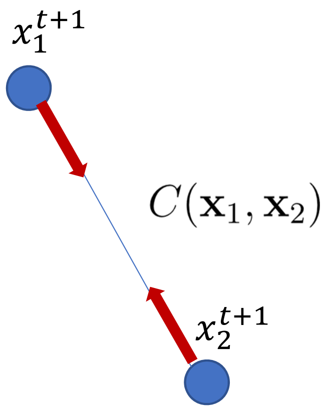
$$\text{s.t. } C(x) + \frac{\alpha}{h^2}\lambda = 0$$

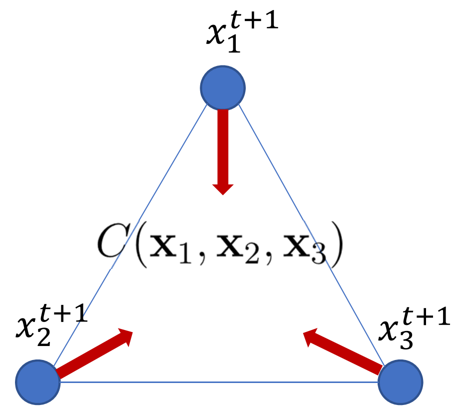


Example Constraints

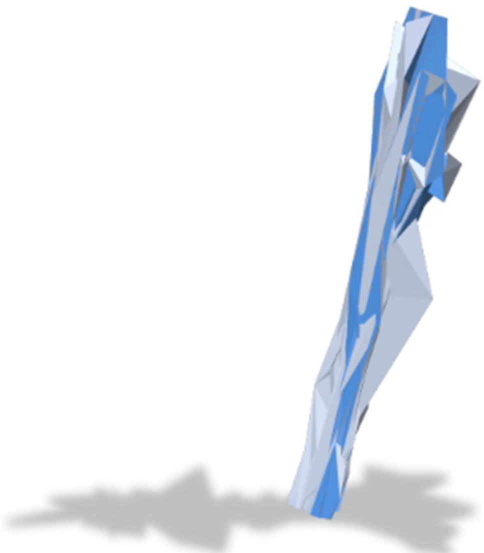
$$C(\mathbf{x}_i, \mathbf{x}_j) = k\left(\|\mathbf{x}_i - \mathbf{x}_j\|_2 - d\right)$$

$$C(\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k) = \frac{1}{2}|(\mathbf{x}_j - \mathbf{x}_i) \times (\mathbf{x}_k - \mathbf{x}_i)| - A$$





Neo-Hookean Elastic Constraints



Macklin et al. 2021

Local Nonlinear Constraint Solver

x :=predict positions given f_{ext} ⬅ **Predict positions**

for $k := 1 \dots K$

    for C_j in constraints C

        Δx :=project constraint C_j ⬅ **Solve constraints**

        $x := x + \Delta x$

$x^{t+1} := x$

$$v^{t+1} := \frac{x^{t+1} - x^t}{h}$$

⬅ **Update solution**

Parallelizing the Solver

x :=predict positions given f_{ext}

for $k := 1 \dots K$

  for p in partitions P

    for C_j in partition p **in parallel**

      $\Delta\lambda :=$ lagrange multiplier change

      $\Delta x := M^{-1} \nabla C_j^T(x) \Delta\lambda$

      $x := x + \Delta x$

$x^{t+1} := x$

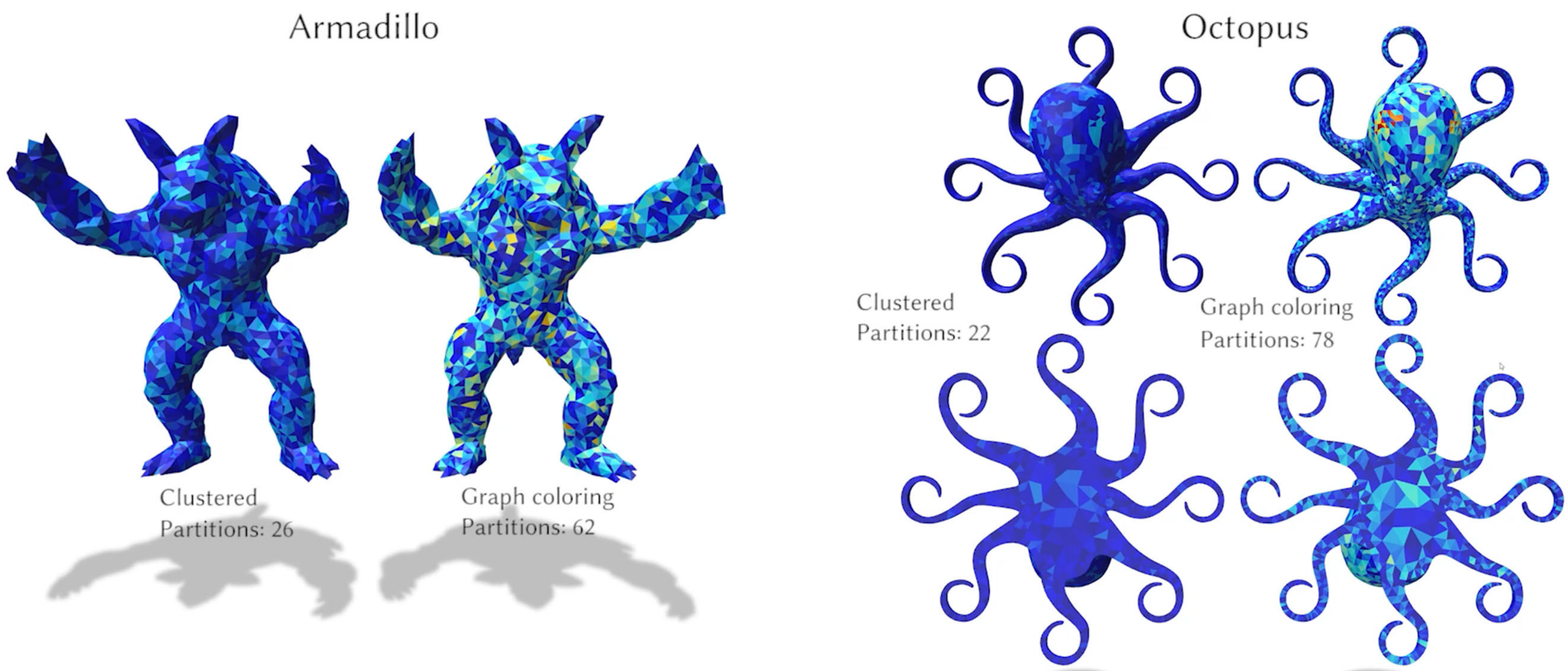
$v^{t+1} := \dfrac{x^{t+1} - x^t}{h}$

Different partitions!

$$\nabla C_i = \begin{bmatrix} \frac{\partial C_j}{\partial x_1} \\ 0 \\ 0 \\ 0 \\ \frac{\partial C_i}{\partial x_6} \\ \frac{\partial C_i}{\partial x_7} \end{bmatrix} \qquad \begin{bmatrix} \frac{\partial C_j}{\partial x_1} \\ \frac{\partial C_j}{\partial x_2} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \nabla C_j$$

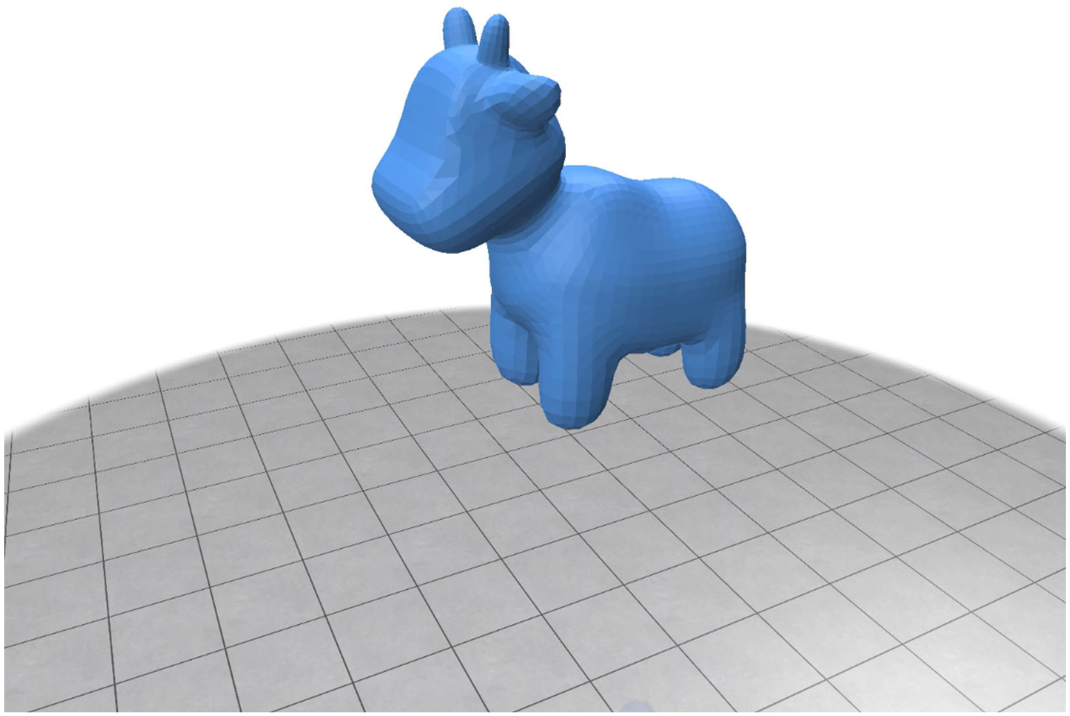
$$\nabla C_i = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \frac{\partial C_i}{\partial x_6} \\ \frac{\partial C_i}{\partial x_7} \end{bmatrix} \qquad \begin{bmatrix} \frac{\partial C_j}{\partial x_1} \\ \frac{\partial C_j}{\partial x_2} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \nabla C_j$$

Contribution #1: Graph Clustering Parallelism

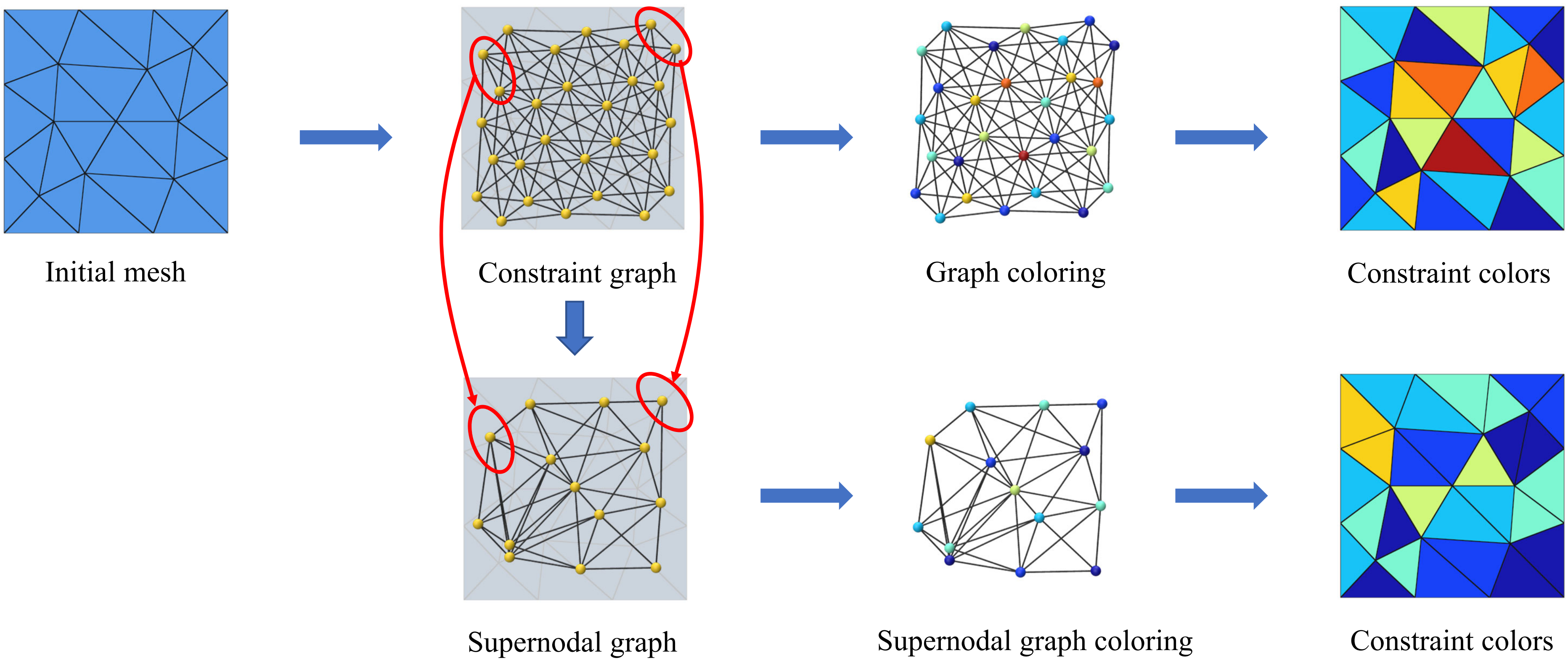


Contribution #2: Elastic Constraint Coupling

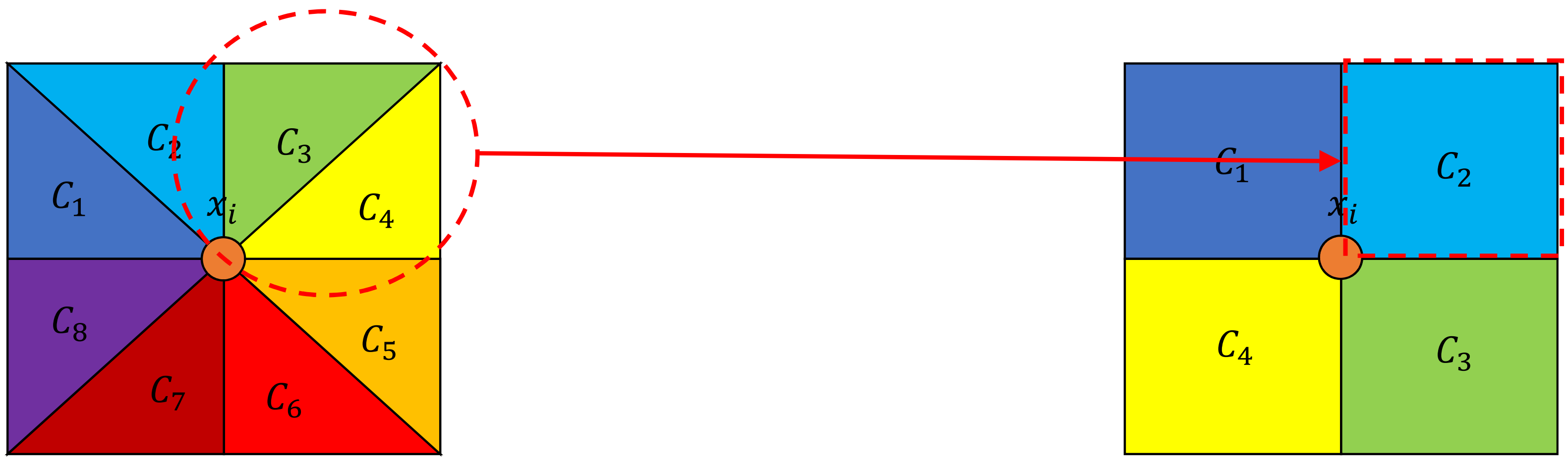
Challenge #1: Enhance Parallelism



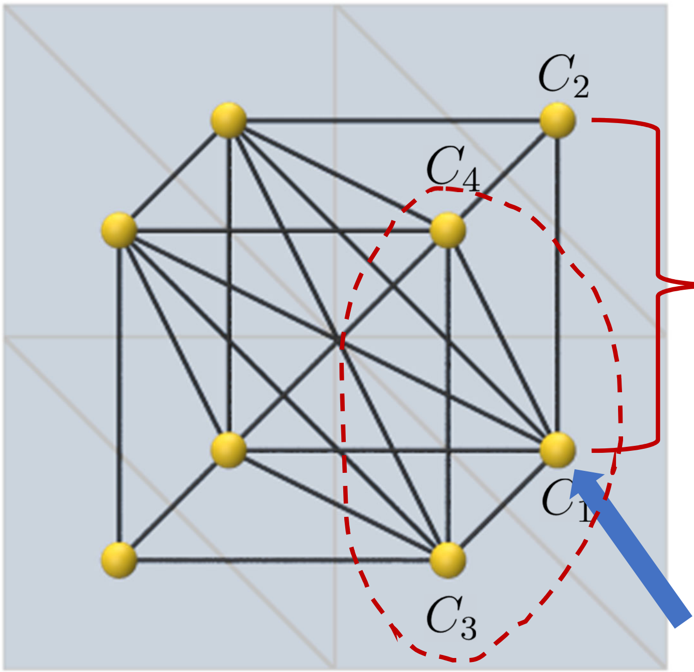
Graph Clustering Solution



Triangle Meshes vs. Quadrilateral Meshes



Our Greedy Graph Clustering

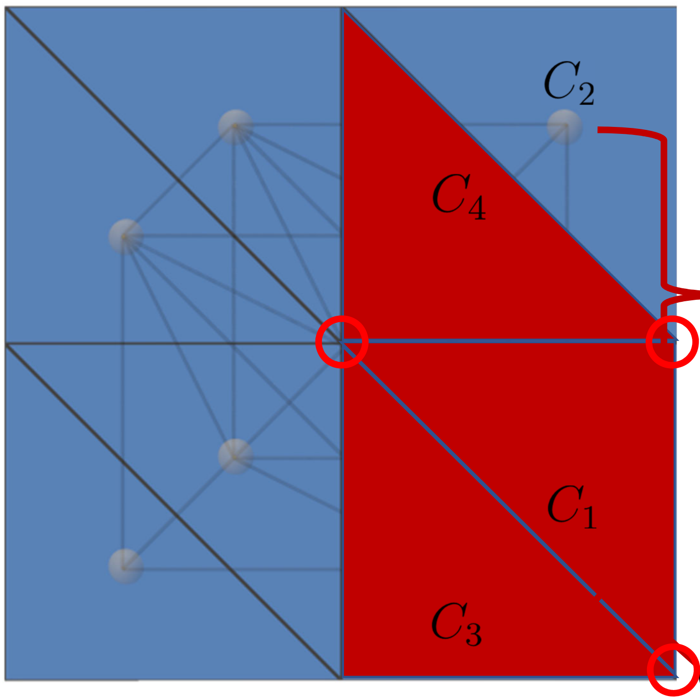


Distance **decreases** as number of shared degrees of freedom **increases**

$$d(C_i, C_j) = 1 - \frac{\mathrm{dofs}(C_i) \cap \mathrm{dofs}(C_j)}{\mathrm{dofs}(C_i) \cup \mathrm{dofs}(C_j)}$$

Breadth First Search

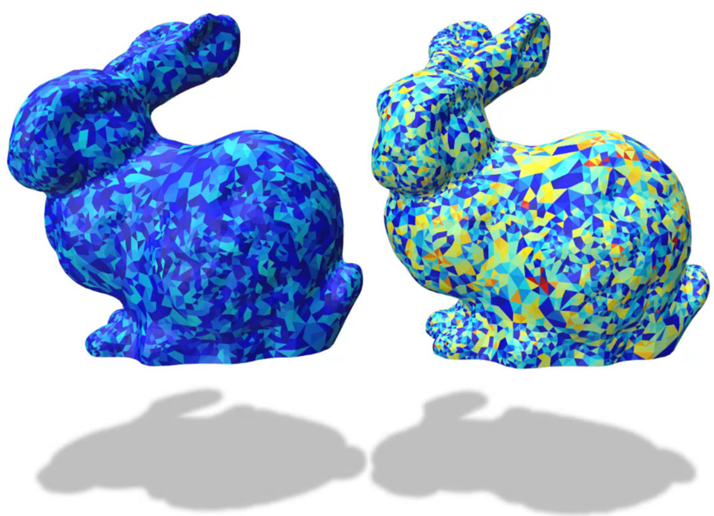
Our Greedy Graph Clustering



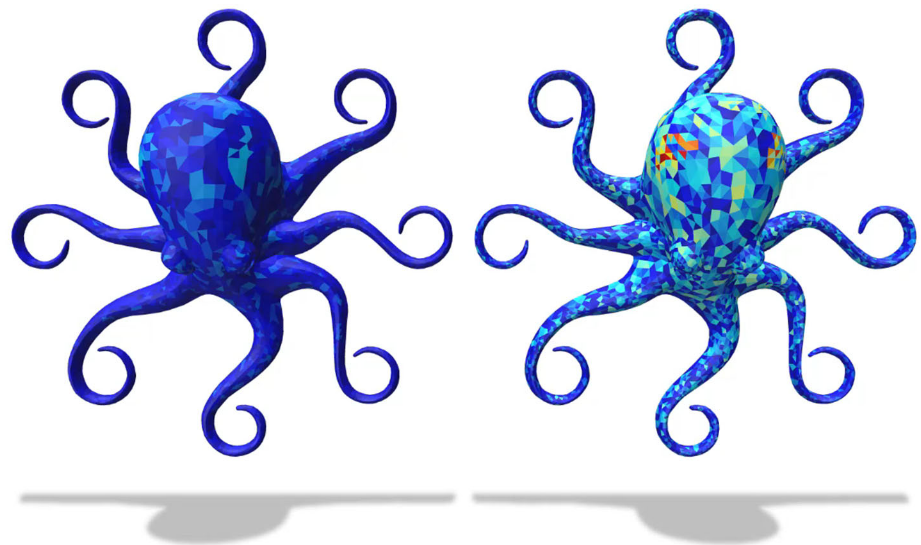
$$d(C_1, C_2) = 1 - \frac{2}{10} = \frac{4}{5}$$

$$d(C_1, C_3) = 1 - \frac{4}{8} = \frac{1}{2}$$

Results

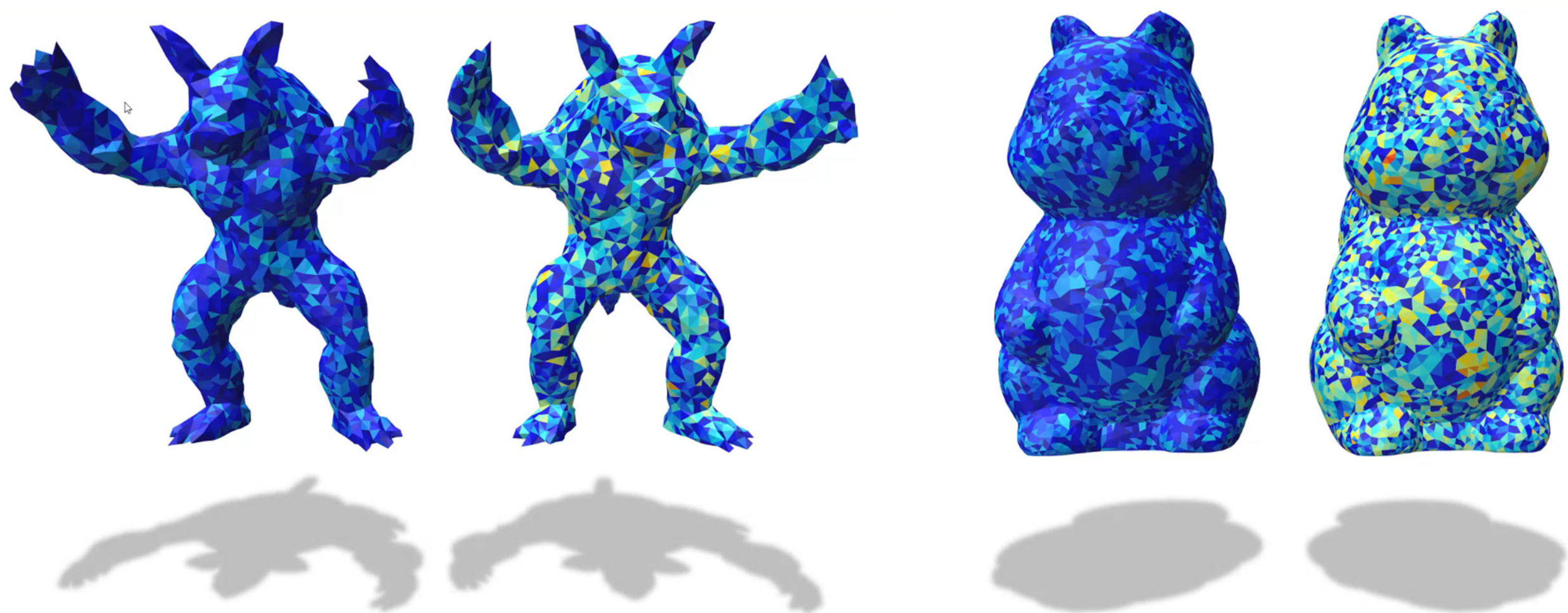


21 colors 52 colors



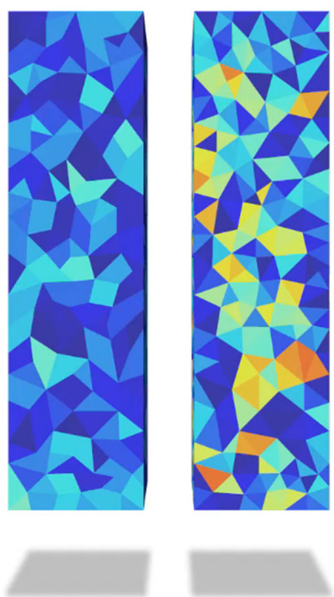
22 colors 78 colors

Results



26 colors 62 colors 22 colors 56 colors

Results



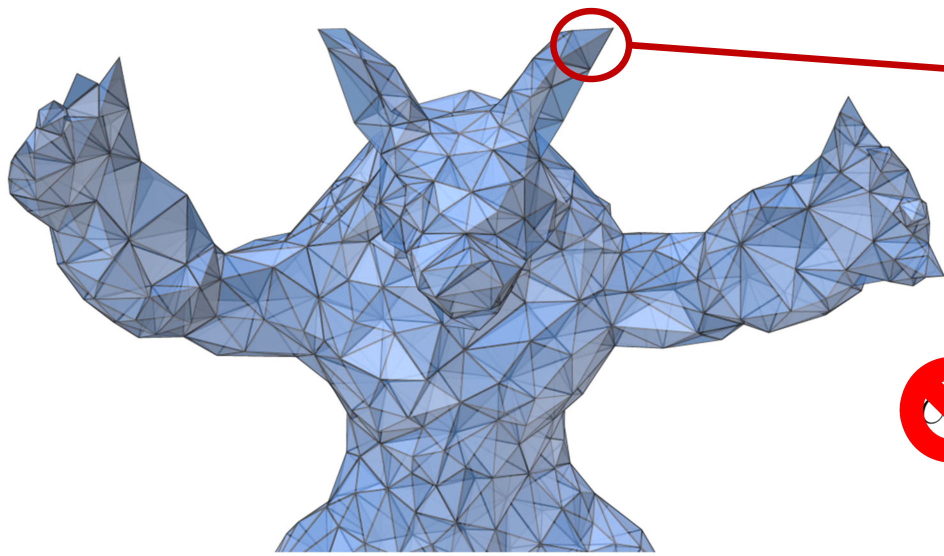
17 colors 40 colors

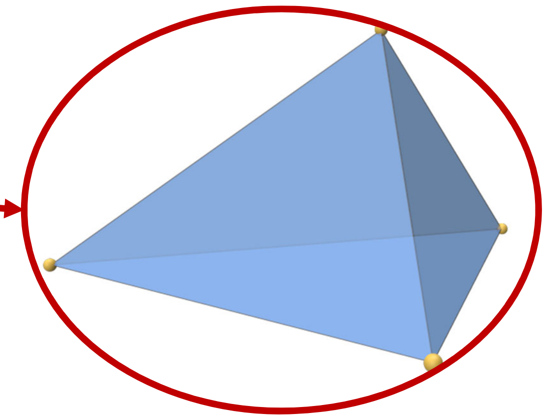
25 colors 104 colors

Challenge #2: Enhance Solver Convergence

30 iterations insufficient!

Neo-Hookean Constraint Formulation





$$C_e(\mathbf{x}) = \mathrm{vol}(\Omega^e)\left[\underbrace{\frac{\lambda}{2}(\det(\mathbf{F})-\gamma)^2}_{\Psi_H(\mathbf{F})} + \underbrace{\frac{\mu}{2}(\mathrm{tr}(\mathbf{F}^T\mathbf{F})-3)}_{\Psi_D(\mathbf{F})}\right]$$

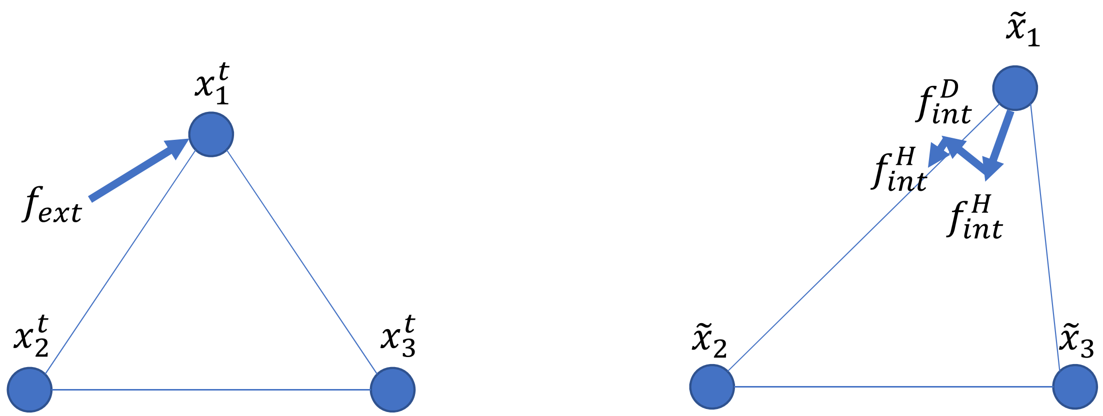
Macklin et al. 2021

$$C_e^H(\mathbf{x}) = \det(\mathbf{F}) - \gamma$$

$$C_e^D(\mathbf{x}) = \sqrt{\mathrm{tr}(\mathbf{F}^T\mathbf{F}) - 3}$$

Decoupled Constraint Formulation

$$\nabla C^H(\mathbf{x}) = -\mu \mathbf{I} : \frac{\partial \mathbf{F}}{\partial x} \neq 0$$
$$\nabla C^D(\mathbf{x}) = \mu \mathbf{I} : \frac{\partial \mathbf{F}}{\partial x} \neq 0$$



Our Coupling Approach

$$C^{\text{neo}}(\mathbf{x}) = \begin{bmatrix} C^H(\mathbf{x}) \\ C^D(\mathbf{x}) \end{bmatrix}$$

$$\nabla C^{\text{neo}}(\mathbf{x}) = \begin{bmatrix} \nabla C^H(\mathbf{x}) \\ \nabla C^D(\mathbf{x}) \end{bmatrix}$$

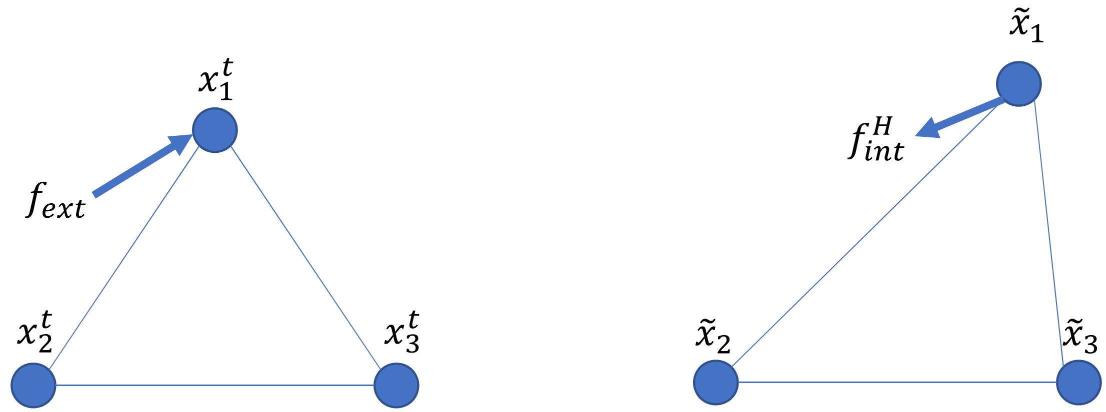
$$\|\nabla C^{\text{neo}}(\mathbf{x})\|_{\text{F}} = 0$$

$$\text{dofs}(C^{\text{neo}}) = \text{dofs}(C^H) \cup \text{dofs}(C^D)$$

Updated Constraint Projection

$$\Delta\lambda_j = -A^{-1}_{2\times 2}\, b_{2\times 1}$$

$$\Delta \mathbf{x} = \mathbf{M}^{-1}\nabla C^{\mathrm{neo}}_{2\times \mathrm{n}}(\mathbf{x})^T \Delta\lambda_{j\,2\times 1}$$



Results

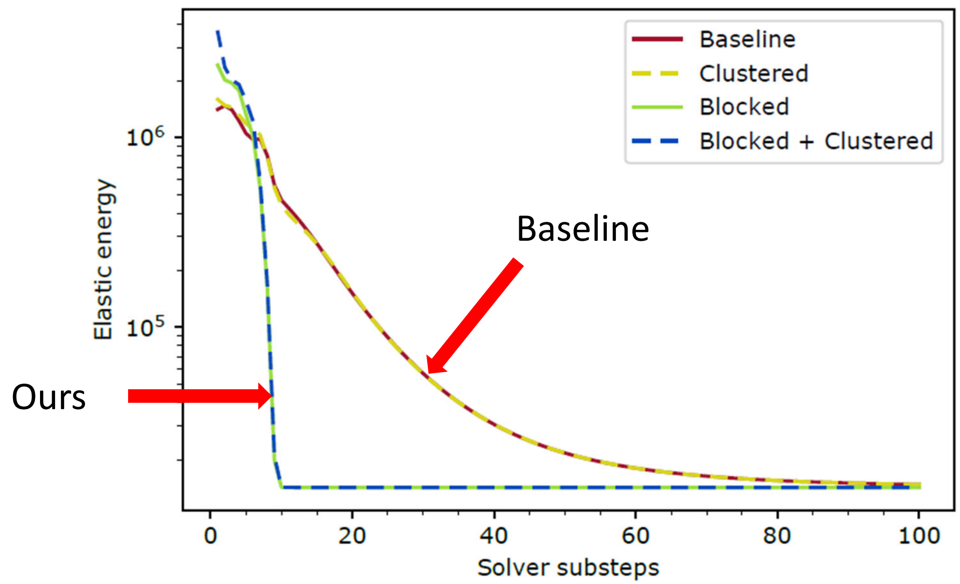
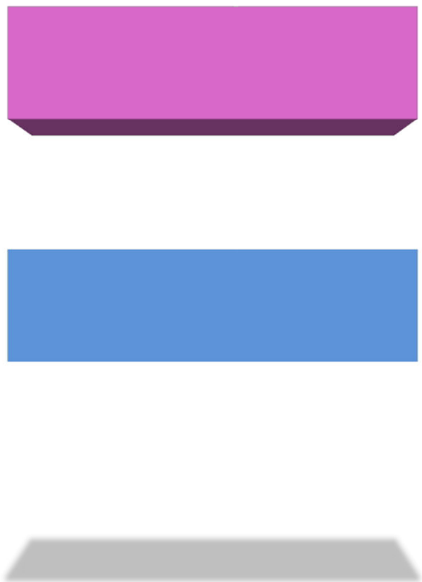


Fig. 5: Convergence for the Beam example using different GPU solver variants shows the superior behavior of block solves of coupled neo-Hookean constraints, while clustering does not hinder convergence.

Results



3.7k tetrahedra
30 sub-steps
$Y = 10$ Mpa
$\nu = 0.45$

2.9k tetrahedra
20 sub-steps
$Y = 10$ Mpa
$\nu = 0.45$

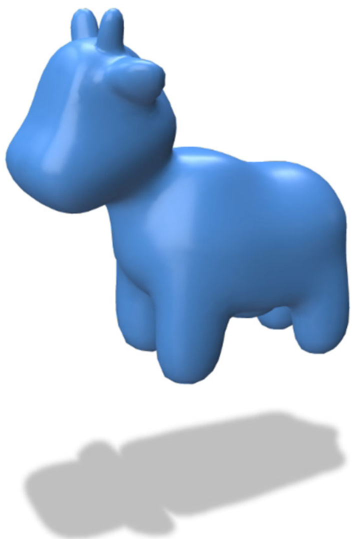
Results

Table 2: Timing results using constraint blocking and clustering. The abbreviations used in column headings are C for clustered and B for blocked constraints. Timings are reported in milliseconds. Speedups are reported with respect to the baseline timings.

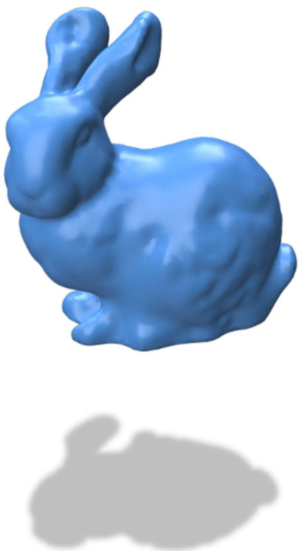
Model	Tetrahedra	Colors			Time (ms)				Speedup		
		Baseline	C	Reduction	Baseline	C	B	B+C	C	B	B+C
Beam	3727	40	17	0.43	285.03	170.49	41.29	20.46	1.67	6.90	13.93
Bunny	56371	52	21	0.40	432.28	315.33	90.86	52.81	1.37	4.76	8.19
Armadillo	45593	62	26	0.42	528.26	374.81	108.89	61.13	1.41	4.85	8.64
Spot	19835	104	25	0.24	885.07	324.18	173.78	54.25	2.73	5.09	16.31
Octopus	22213	78	22	0.28	663.96	299.78	132.76	49.10	2.21	5.00	13.52
Squirrel	64768	56	22	0.39	457.91	332.16	94.78	55.41	1.38	4.83	8.26

Results

162k tetrahedra
50 sub-steps
$Y = 400$ Mpa
$\nu = 0.499$



19.8k tetrahedra
50 sub-steps
$Y = 30$ Mpa
$\nu = 0.45$



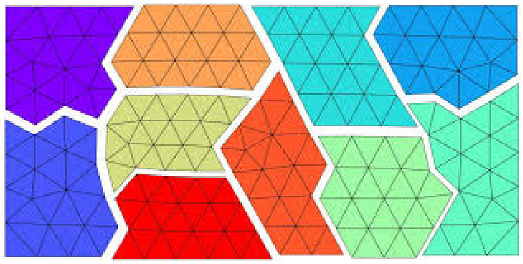
56.4k tetrahedra
50 sub-steps
$Y = 20$ Mpa
$\nu = 0.45$

Conclusion

- **Better parallelism**
  - Isolates dense sub-graphs using graph clustering
- **Faster convergence**
  - Project constraint blocks by solving small linear systems
- Methods are ***general*** and ***overhead-free***

Future Work

- Sophisticated graph clustering/partitioning strategies
- Judicious constraint coupling strategies
- Dynamic updates to clusters/partitions



Kong, F., Stogner, S. T. et al. 2018

Acknowledgements

symgery

NSERC
CRSNG

Thank You!