Robust Treatment of Degenerate Elements in Interactive Corotational FEM Simulations

O. Civit-Flores and A. Susín

UPC-BarcelonaTech

June 11, 2014

Outline

Interactive FEM

Interactive simulation of deformable solids using FEM

Applications:

- Virtual reality, surgery, training...
- Videogames

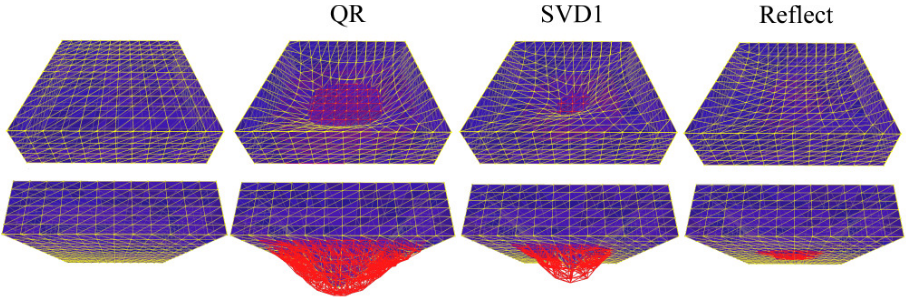
Requirements:

- User interaction
- Efficiency
- Robustness
- Realism

Contribution

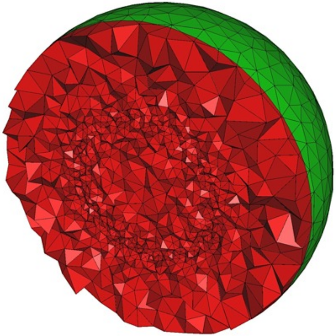
Element degeneration threatens robustness and realism:

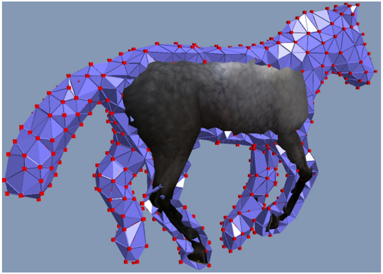
- We identify issues with existing degenerate element treatment schemes
- We propose a new method that avoids them



Outline

Partition the computational domain Ω into sub-domains Ω_i with N shared nodes





Linear FEM

Tetrahedral elements:

$$\boldsymbol{r}_e = \begin{bmatrix} \boldsymbol{r}_1 \\ \vdots \\ \boldsymbol{r}_4 \end{bmatrix}, \quad \boldsymbol{x}_e = \begin{bmatrix} \boldsymbol{x}_1 \\ \vdots \\ \boldsymbol{x}_4 \end{bmatrix}, \quad \boldsymbol{f}_e = \begin{bmatrix} \boldsymbol{f}_1 \\ \vdots \\ \boldsymbol{f}_4 \end{bmatrix}$$

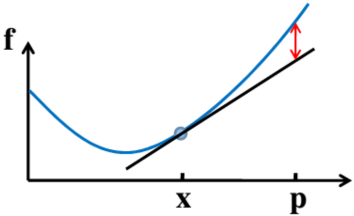
The elastic forces on the nodes are:

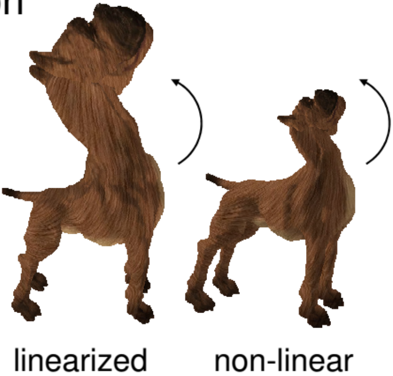
$$\boldsymbol{f}_e = -\mathcal{K}_e \boldsymbol{u}_e, \quad \boldsymbol{u}_e = \boldsymbol{x}_e - \boldsymbol{r}_e$$

Properties:

- Constant stiffness matrix $\mathcal{K}_e$
- Invariant to translation, but not to rotation

Linearization only valid close to the point of linearization

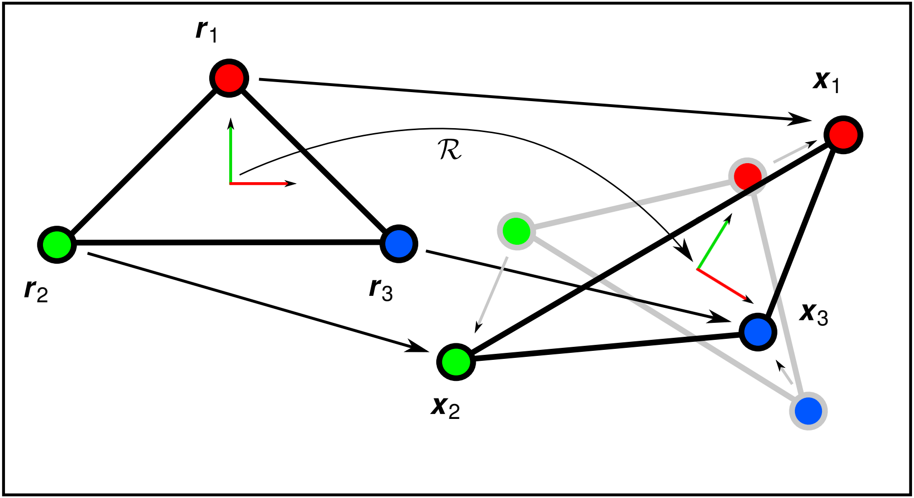




From [1]

Corotational Linear FEM (I)

Idea: Apply LFEM in a reference system local to each element



Corotational Linear FEM (II)

1. Compute deformation matrix $\mathcal{F}$:

$$\mathcal{F} = \mathcal{D}(\boldsymbol{x}_e)\mathcal{D}(\boldsymbol{r}_e)^{-1}, \quad \mathcal{D}(\boldsymbol{v}_e) = \begin{bmatrix} \boldsymbol{v}_2 - \boldsymbol{v}_1 & \boldsymbol{v}_3 - \boldsymbol{v}_1 & \boldsymbol{v}_4 - \boldsymbol{v}_1 \end{bmatrix}$$

2. Factorize into rotation and scaling:

$$\mathcal{F} = \mathcal{R}\mathcal{S}$$

3. Apply linear elasticity in local coordinates:

$$\boldsymbol{f}_e = -\mathcal{R}_e\mathcal{K}_e(\mathcal{R}_e^T\boldsymbol{x}_e - \boldsymbol{r}_e)$$

Properties:

- Geometrically non-linear
- Invariant to translation and rotation

Corotational Linear FEM (II)

1. Compute deformation matrix $\mathcal{F}$:

$$\mathcal{F} = \mathcal{D}(\boldsymbol{x}_e)\mathcal{D}(\boldsymbol{r}_e)^{-1}, \quad \mathcal{D}(\boldsymbol{v}_e) = \begin{bmatrix} \boldsymbol{v}_2 - \boldsymbol{v}_1 & \boldsymbol{v}_3 - \boldsymbol{v}_1 & \boldsymbol{v}_4 - \boldsymbol{v}_1 \end{bmatrix}$$

2. Factorize into rotation and scaling:

$$\mathcal{F} = \mathcal{R}\mathcal{S}$$

3. Apply linear elasticity in local coordinates:

$$\boldsymbol{f}_e = -\mathcal{R}_e\mathcal{K}_e(\mathcal{R}_e^T\boldsymbol{x}_e - \boldsymbol{r}_e)$$

Properties:

- Geometrically non-linear
- Invariant to translation and rotation

Corotational Linear FEM (II)

1. Compute deformation matrix $\mathcal{F}$:

$$\mathcal{F} = \mathcal{D}(\boldsymbol{x}_e)\mathcal{D}(\boldsymbol{r}_e)^{-1}, \quad \mathcal{D}(\boldsymbol{v}_e) = \begin{bmatrix} \boldsymbol{v}_2 - \boldsymbol{v}_1 & \boldsymbol{v}_3 - \boldsymbol{v}_1 & \boldsymbol{v}_4 - \boldsymbol{v}_1 \end{bmatrix}$$

2. Factorize into rotation and scaling:

$$\mathcal{F} = \mathcal{R}\mathcal{S}$$

3. Apply linear elasticity in local coordinates:

$$\boldsymbol{f}_e = -\mathcal{R}_e\mathcal{K}_e(\mathcal{R}_e^T\boldsymbol{x}_e - \boldsymbol{r}_e)$$

Properties:

- Geometrically non-linear
- Invariant to translation and rotation

Corotational Linear FEM (II)

1. Compute deformation matrix $\mathcal{F}$:

$$\mathcal{F} = \mathcal{D}(\boldsymbol{x}_e)\mathcal{D}(\boldsymbol{r}_e)^{-1}, \quad \mathcal{D}(\boldsymbol{v}_e) = \begin{bmatrix} \boldsymbol{v}_2 - \boldsymbol{v}_1 & \boldsymbol{v}_3 - \boldsymbol{v}_1 & \boldsymbol{v}_4 - \boldsymbol{v}_1 \end{bmatrix}$$

2. Factorize into rotation and scaling:

$$\mathcal{F} = \mathcal{R}\mathcal{S}$$

3. Apply linear elasticity in local coordinates:

$$\boldsymbol{f}_e = -\mathcal{R}_e\mathcal{K}_e(\mathcal{R}_e^T\boldsymbol{x}_e - \boldsymbol{r}_e)$$

Properties:

- Geometrically non-linear
- Invariant to translation and rotation

Corotational Linear FEM (II)

1. Compute deformation matrix $\mathcal{F}$:

$$\mathcal{F} = \mathcal{D}(\boldsymbol{x}_e)\mathcal{D}(\boldsymbol{r}_e)^{-1}, \quad \mathcal{D}(\boldsymbol{v}_e) = \begin{bmatrix} \boldsymbol{v}_2 - \boldsymbol{v}_1 & \boldsymbol{v}_3 - \boldsymbol{v}_1 & \boldsymbol{v}_4 - \boldsymbol{v}_1 \end{bmatrix}$$

2. Factorize into rotation and scaling:

$$\mathcal{F} = \mathcal{R}\mathcal{S}$$

3. Apply linear elasticity in local coordinates:

$$\boldsymbol{f}_e = -\mathcal{R}_e\mathcal{K}_e(\mathcal{R}_e^T\boldsymbol{x}_e - \boldsymbol{r}_e)$$

Properties:

- Geometrically non-linear
- Invariant to translation and rotation

Dynamics and Quasistatics

Node positions $\boldsymbol{x} \in \mathbb{R}^{3N}$ are the DOF:

- Dynamics:

$$\mathcal{M}\ddot{\boldsymbol{x}} = \boldsymbol{f}_s(\boldsymbol{x}) + \boldsymbol{f}_d(\boldsymbol{x}, \dot{\boldsymbol{x}}) + \boldsymbol{f}_{\text{ext}}$$

- Quasistatics:

$$\boldsymbol{f}_s(\boldsymbol{x}) = -\boldsymbol{f}_{\text{ext}}$$

Element Degeneration

Collapse with $|\det(\mathcal{F})| < \epsilon$ or inversion with $\det(\mathcal{F}) < 0$

- Unphysical
- Unavoidable with (finite) linear forces
- Unavoidable due to discretization
- Unavoidable due to user interaction
- $\det(\mathcal{F}) < \epsilon$ affects $\mathcal{F} = \mathcal{R}\mathcal{S}$ factorization

Element Degeneration

Collapse with $|\det(\mathcal{F})| < \epsilon$ or inversion with $\det(\mathcal{F}) < 0$

- Unphysical
- Unavoidable with (finite) linear forces
- Unavoidable due to discretization
- Unavoidable due to user interaction
- $\det(\mathcal{F}) < \epsilon$ affects $\mathcal{F} = \mathcal{R}\mathcal{S}$ factorization

Element Degeneration

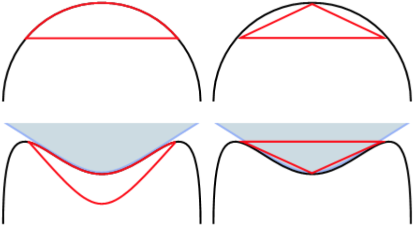
Collapse with $|\det(\mathcal{F})| < \epsilon$ or inversion with $\det(\mathcal{F}) < 0$

- Unphysical
- Unavoidable with (finite) linear forces
- Unavoidable due to discretization
- Unavoidable due to user interaction
- $\det(\mathcal{F}) < \epsilon$ affects $\mathcal{F} = \mathcal{R}\mathcal{S}$ factorization

Element Degeneration

Collapse with $|\det(\mathcal{F})| < \epsilon$ or inversion with $\det(\mathcal{F}) < 0$

- Unphysical
- Unavoidable with (finite) linear forces
- Unavoidable due to discretization
- Unavoidable due to user interaction
- $\det(\mathcal{F}) < \epsilon$ affects $\mathcal{F} = \mathcal{R}\mathcal{S}$ factorization



From [2]

Element Degeneration

Collapse with $|\det(\mathcal{F})| < \epsilon$ or inversion with $\det(\mathcal{F}) < 0$

- Unphysical
- Unavoidable with (finite) linear forces
- Unavoidable due to discretization
- Unavoidable due to user interaction
- $\det(\mathcal{F}) < \epsilon$ affects $\mathcal{F} = \mathcal{R}\mathcal{S}$ factorization

Element Degeneration

Collapse with $|\det(\mathcal{F})| < \epsilon$ or inversion with $\det(\mathcal{F}) < 0$

- Unphysical
- Unavoidable with (finite) linear forces
- Unavoidable due to discretization
- Unavoidable due to user interaction
- $\det(\mathcal{F}) < \epsilon$ affects $\mathcal{F} = \mathcal{RS}$ factorization

Outline

Methods

Several methods to extract $\mathcal{R}$ from $\mathcal{F}$ are possible:

- Polar Decomposition [1]
- QR Factorization [3]
- Hybrid PD-QR [5]
- Modified Singular Value Decomposition (SVD1) [2]
- Coherent Singular Value Decomposition (SVD2) [4]
- Project/Reflect [6]
- Degeneration-Aware Polar Decomposition [**?**]

Polar Decomposition

Factorizes $\mathcal{F} = \mathcal{R}\mathcal{S}$, where $\mathcal{R}$ is orthonormal and $\mathcal{S}$ is symmetric

- Best matching, minimizes $\|\mathcal{F} - \mathcal{R}\|_F^2$
- Fails if $|\det(\mathcal{F})| \leq \epsilon$ (collapsed)
- Reflected $\mathcal{R}$ with $\det(\mathcal{R}) = -1$ if $\det(\mathcal{F}) < 0$ (inverted)

Polar Decomposition

Factorizes $\mathcal{F} = \mathcal{R}\mathcal{S}$, where $\mathcal{R}$ is orthonormal and $\mathcal{S}$ is symmetric

- Best matching, minimizes $\|\mathcal{F} - \mathcal{R}\|_F^2$
- Fails if $|\det(\mathcal{F})| \leq \epsilon$ (collapsed)
- Reflected $\mathcal{R}$ with $\det(\mathcal{R}) = -1$ if $\det(\mathcal{F}) < 0$ (inverted)

Polar Decomposition

Factorizes $\mathcal{F} = \mathcal{R}\mathcal{S}$, where $\mathcal{R}$ is orthonormal and $\mathcal{S}$ is symmetric

- Best matching, minimizes $\|\mathcal{F} - \mathcal{R}\|_F^2$
- Fails if $|\det(\mathcal{F})| \leq \epsilon$ (collapsed)
- Reflected $\mathcal{R}$ with $\det(\mathcal{R}) = -1$ if $\det(\mathcal{F}) < 0$ (inverted)

QR Factorization

Factorizes $\mathcal{F} = \mathcal{R}\mathcal{E}$ using Gram-Schmidt orthonormalization, where $\mathcal{R}$ is orthonormal and $\mathcal{E}$ is upper-triangular

- Fast and Robust
- Handles collapsed and inverted elements seamlessly
- Induces Anisotropy
- Critical point on collapse plane

QR Factorization

Factorizes $\mathcal{F} = \mathcal{R}\mathcal{E}$ using Gram-Schmidt orthonormalization, where $\mathcal{R}$ is orthonormal and $\mathcal{E}$ is upper-triangular

- Fast and Robust
- Handles collapsed and inverted elements seamlessly
- Induces Anisotropy
- Critical point on collapse plane

QR Factorization

Factorizes $\mathcal{F} = \mathcal{R}\mathcal{E}$ using Gram-Schmidt orthonormalization, where $\mathcal{R}$ is orthonormal and $\mathcal{E}$ is upper-triangular

- Fast and Robust
- Handles collapsed and inverted elements seamlessly
- Induces Anisotropy
- Critical point on collapse plane

Hybrid PD-QR

Use Polar Decomposition for undegenerate elements and QR for degenerate ones below a threshold $\det(\mathcal{F}) < \alpha$

- Inherits good properties of PD and QR...
- ...but also the drawbacks of QR...
- ...and adds discontinuity across transition

Hybrid PD-QR

Use Polar Decomposition for undegenerate elements and QR for degenerate ones below a threshold $\det(\mathcal{F}) < \alpha$

- Inherits good properties of PD and QR...
- ...but also the drawbacks of QR...
- ...and adds discontinuity across transition

Hybrid PD-QR

Use Polar Decomposition for undegenerate elements and QR for degenerate ones below a threshold $\det(\mathcal{F}) < \alpha$

- Inherits good properties of PD and QR...
- ...but also the drawbacks of QR...
- ...and adds discontinuity across transition

Modified Singular Value Decomposition (SVD1)

SVD factorizes $\mathcal{F} = \mathcal{U}\hat{\mathcal{F}}\mathcal{V}^T$, where $\mathcal{U}$ and $\mathcal{V}^T$ are orthonormal and $\hat{\mathcal{F}}$ is diagonal, with singular values $\sigma_i \geq \sigma_{i+1} \geq 0$

- Handles collapsed elements
- Handles inverted elements negating the smallest σ_i
- $\bar{\mathcal{R}} = \bar{\mathcal{U}}\bar{\mathcal{V}}^T$ equivalent to “invertible” PD
- Computationally expensive
- Critical point at repeated singular values

Modified Singular Value Decomposition (SVD1)

SVD factorizes $\mathcal{F} = \mathcal{U}\hat{\mathcal{F}}\mathcal{V}^T$, where $\mathcal{U}$ and $\mathcal{V}^T$ are orthonormal and $\hat{\mathcal{F}}$ is diagonal, with singular values $\sigma_i \geq \sigma_{i+1} \geq 0$

- Handles collapsed elements
- Handles inverted elements negating the smallest σ_i
- $\bar{\mathcal{R}} = \bar{\mathcal{U}}\bar{\mathcal{V}}^T$ equivalent to "invertible" PD
- Computationally expensive
- Critical point at repeated singular values

Modified Singular Value Decomposition (SVD1)

SVD factorizes $\mathcal{F} = \mathcal{U}\hat{\mathcal{F}}\mathcal{V}^T$, where $\mathcal{U}$ and $\mathcal{V}^T$ are orthonormal and $\hat{\mathcal{F}}$ is diagonal, with singular values $\sigma_i \geq \sigma_{i+1} \geq 0$

- Handles collapsed elements
- Handles inverted elements negating the smallest σ_i
- $\bar{\mathcal{R}} = \bar{\mathcal{U}}\bar{\mathcal{V}}^T$ equivalent to “invertible” PD
- Computationally expensive
- Critical point at repeated singular values

Coherent Singular Value Decomposition (SVD2)

Tries to improve SVD1 by enforcing temporal coherence in the inferred inversion direction (smallest σ_i)

- Better aligned forces
- Computationally expensive
- Discontinuous

Coherent Singular Value Decomposition (SVD2)

Tries to improve SVD1 by enforcing temporal coherence in the inferred inversion direction (smallest σ_i)

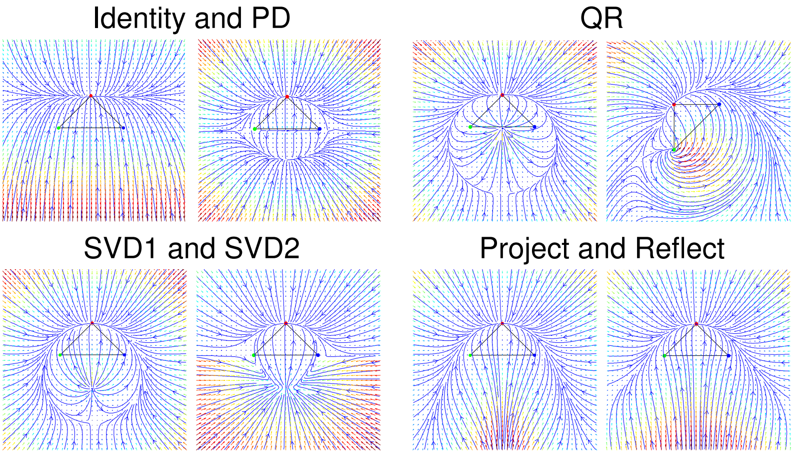
- Better aligned forces
- Computationally expensive
- Discontinuous

Coherent Singular Value Decomposition (SVD2)

Tries to improve SVD1 by enforcing temporal coherence in the inferred inversion direction (smallest σ_i)

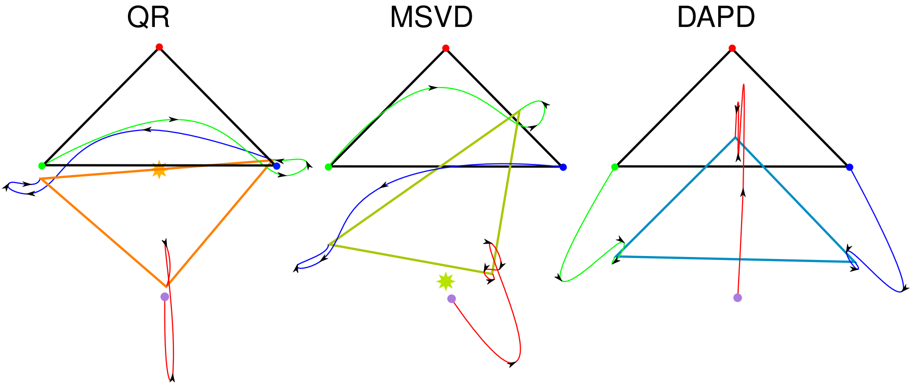
- Better aligned forces
- Computationally expensive
- Discontinuous

Comparison: Force Fields



Forcefields experienced by $\boldsymbol{x}_1$ (red) by in a $[-3, 3]^2$ region when $\boldsymbol{x}_2, \boldsymbol{x}_3$ (green, blue) are fixed

Comparison: Trajectories



Discussion

The main problems are:

- Discrete-time heuristic ignores trajectories: QR, SVD1, SVD2
- Critical points induce counter-intuitive rotations: QR, SVD1
- Discontinuity induces jitter: Hybrid PD-QR, SVD2

Our solution: Degeneration-Aware Polar Decomposition

Discussion

The main problems are:

- Discrete-time heuristic ignores trajectories: QR, SVD1, SVD2
- Critical points induce counter-intuitive rotations: QR, SVD1
- Discontinuity induces jitter: Hybrid PD-QR, SVD2

Our solution: Degeneration-Aware Polar Decomposition

Outline

Idea

- Use continuous time to detect collapse
- Compute $\mathcal{R}$ avoiding rotation-past-collapse

Detecting collapse (I)

Given a timestep $[t_0, t_1]$:

1. New degeneration if $\det(\mathcal{F}(t_0)) > \alpha$ and $\det(\mathcal{F}(t_1)) \leq \alpha$
2. Compute $t_c \in [t_0, t_1]$ solving $\det(\mathcal{F}(t_c)) = \alpha$
3. Compute the collapse feature pair (A_c, B_c) at t_c

Details:

- In 2D only V-E case, in 3D V-F and E-E cases
- Degeneration threshold $\alpha > 0$ improves numerical robustness

Detecting collapse (I)

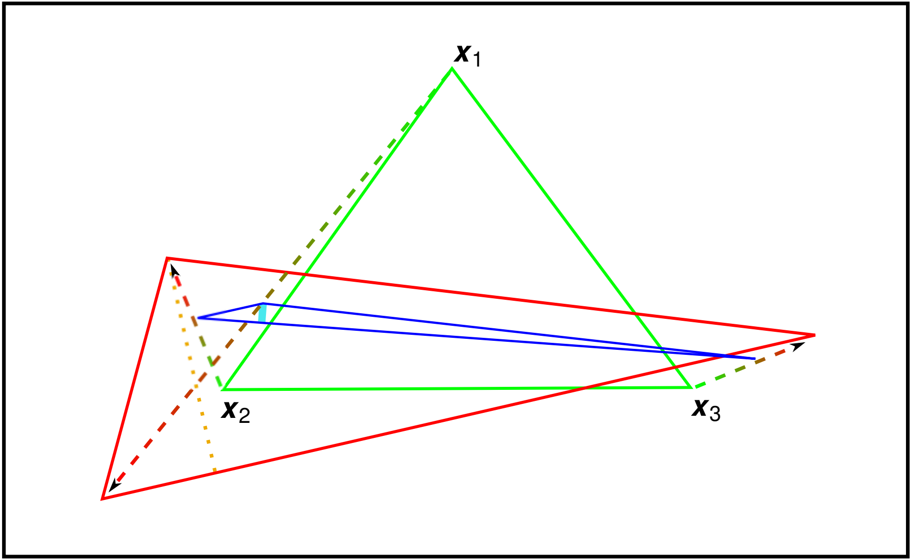
Given a timestep $[t_0, t_1]$:

1. New degeneration if $\det(\mathcal{F}(t_0)) > \alpha$ and $\det(\mathcal{F}(t_1)) \leq \alpha$
2. Compute $t_c \in [t_0, t_1]$ solving $\det(\mathcal{F}(t_c)) = \alpha$
3. Compute the collapse feature pair (A_c, B_c) at t_c

Details:

- In 2D only V-E case, in 3D V-F and E-E cases
- Degeneration threshold $\alpha > 0$ improves numerical robustness

Detecting collapse (II)



Computing $\mathcal{R}$ (I)

We compute $\bar{\mathcal{R}}$ for degenerate elements as follows:

1. Compute degeneration direction $\hat{\boldsymbol{d}}_c$ from (A_c, B_c)
2. Displace nodes of feature A_c as $\bar{\boldsymbol{x}}_i = \boldsymbol{x}_i + \lambda(\alpha, \beta)\hat{\boldsymbol{d}}_c$
3. Compute $\bar{\mathcal{F}}$ from displaced nodes $\bar{\boldsymbol{x}}_i$
4. Compute $\bar{\mathcal{R}}$ using Polar Decomposition on $\bar{\mathcal{F}}$

Details:

- $\lambda(\alpha, \beta)$ computes a displacement length that reverts element degeneration and guarantees $\det(\bar{\mathcal{F}}) > \epsilon$
- Parameter β controls force alignment

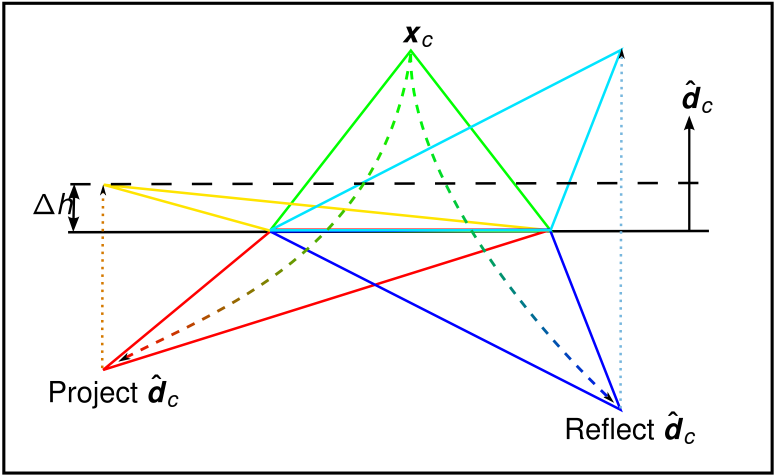
Computing $\mathcal{R}$ (I)

We compute $\bar{\mathcal{R}}$ for degenerate elements as follows:

1. Compute degeneration direction $\hat{\boldsymbol{d}}_c$ from (A_c, B_c)
2. Displace nodes of feature A_c as $\bar{\boldsymbol{x}}_i = \boldsymbol{x}_i + \lambda(\alpha, \beta)\hat{\boldsymbol{d}}_c$
3. Compute $\bar{\mathcal{F}}$ from displaced nodes $\bar{\boldsymbol{x}}_i$
4. Compute $\bar{\mathcal{R}}$ using Polar Decomposition on $\bar{\mathcal{F}}$

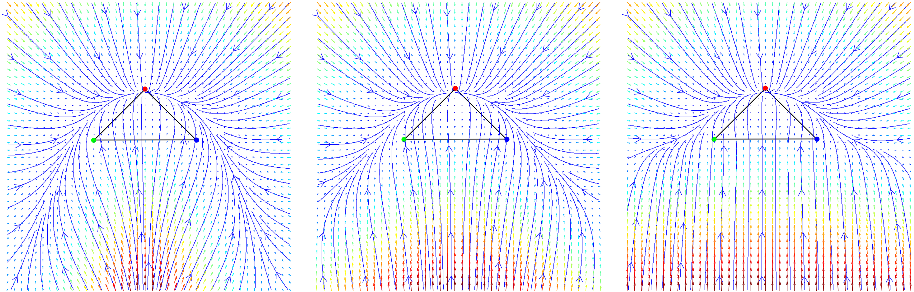
Details:

- $\lambda(\alpha, \beta)$ computes a displacement length that reverts element degeneration and guarantees $\det(\bar{\mathcal{F}}) > \epsilon$
- Parameter β controls force alignment



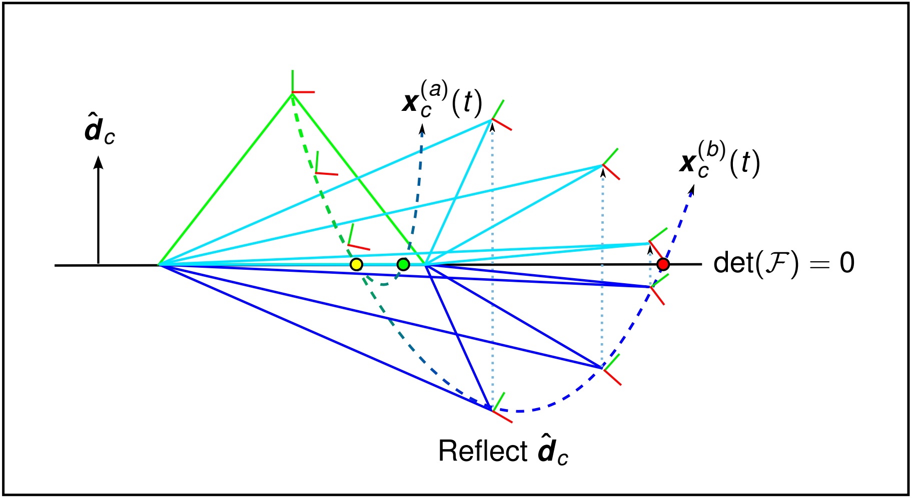
Project ($\beta = 1$) and Reflect ($\beta = 2$). Height Δh depends on α

Computing $\mathcal{R}$ (III)

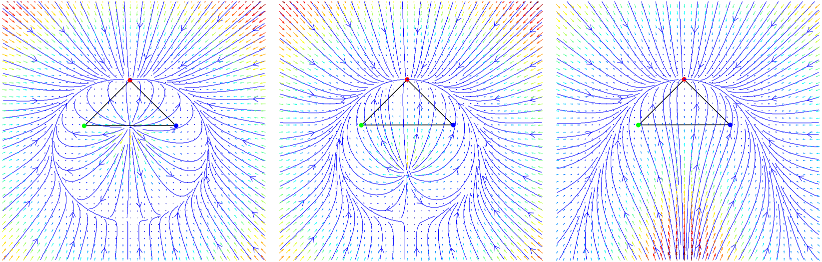


The effect of parameter β for values 1, 2 and 10

Computing $\mathcal{R}$ (VI)



Discussion

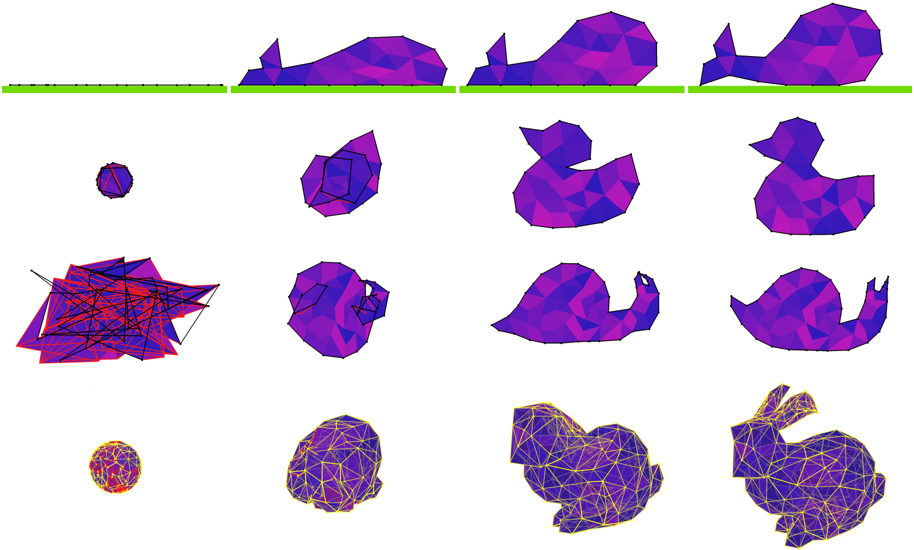


DAPD:

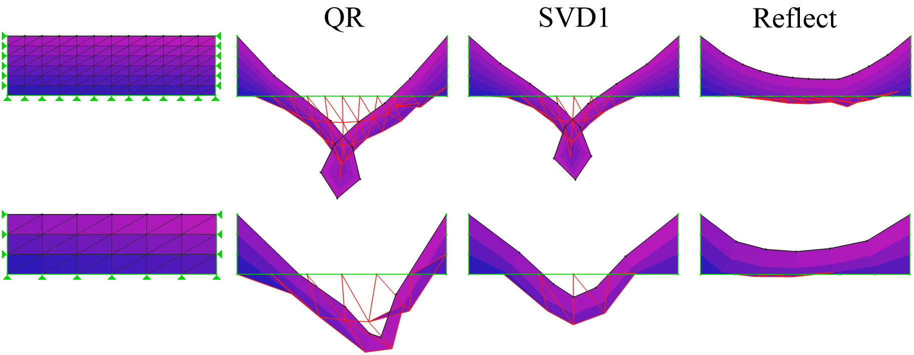
- Cached pair (A_c, B_c) ensures temporal coherency
- Coherently aligned recovery forces
- Continuous and differentiable everywhere...
- ...except when A_c or B_c collapse during inversion

Outline

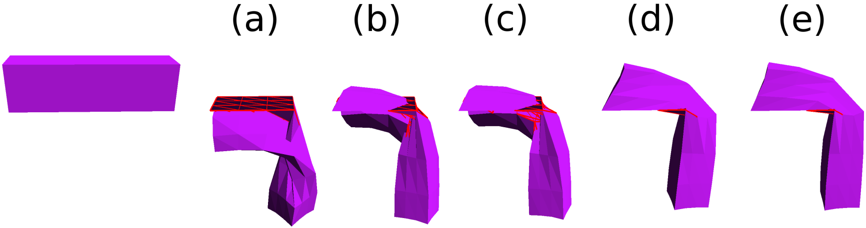
Results



Results



Results



(a) QR, (b) SVD1, (c) DAPD $\beta = 1$, (d) DAPD $\beta = 5$, (e) Nonlinear

Results

`http://www.lsi.upc.edu/~ocivit/videos/DCFEM_Full.ogg`
`http://www.lsi.upc.edu/~ocivit/videos/DCNLFEM-Draft.avi`

Conclusions

Advantages of DAPD:

- Increased realism
- Shorter recovery time
- Faster than SVD1

Limitations:

- Cannot handle initially degenerate elements
- Limited to triangular/tetrahedral elements
- Slower than QR

References

M.Müller, M.Gross
Interactive virtual materials
Proceedings of Graphics Interface 2004

G.Irving, J.Teran, R.Fedkiw
Invertible finite elements for robust simulation of large deformation,
Symposium on Computer Animation 2004

M.Nesme, Y.Payan, F.Faure
Efficient, Physically Plausible Finite Elements
Eurographics 2005

R.Schmedding, M.Teschner
Inversion handling for stable deformable modeling
The Visual Computer 2008

E.Parker, J.O'Brien
Real-time deformation and fracture in a game environment
Symposium on Computer Animation 2009

O. Civit-Flores, A. Susín
Robust treatment of degenerate elements in interactive corotational FEM simulations
(to appear) Computer Graphics Forum 2014

Q&A

?