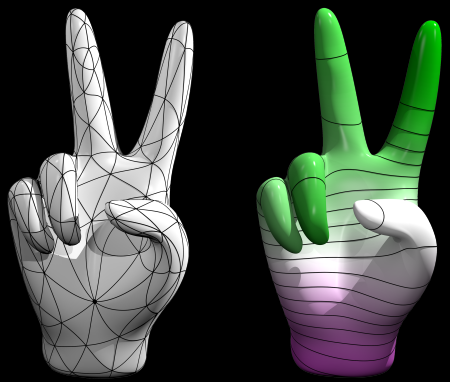
Isogeometric Approximation of Variational Shell Problems

PhD Defense Ricardo Perl

April 12, 2016 Bonn

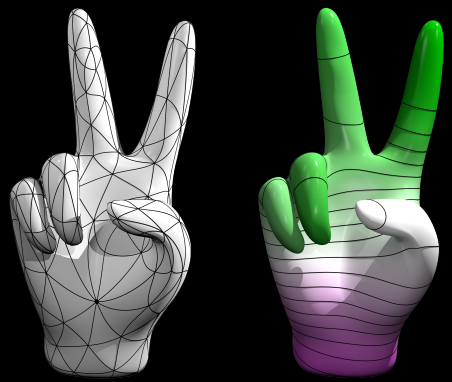
Outlook

Isogeometric
Subdivision Method



Outlook

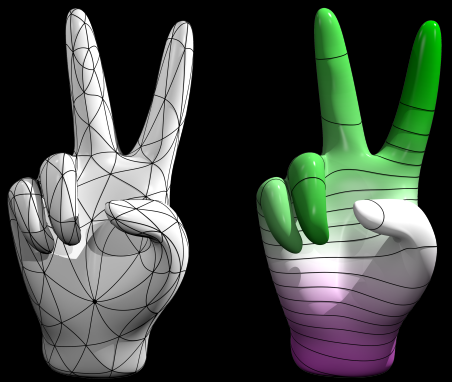
Isogeometric
Subdivision Method



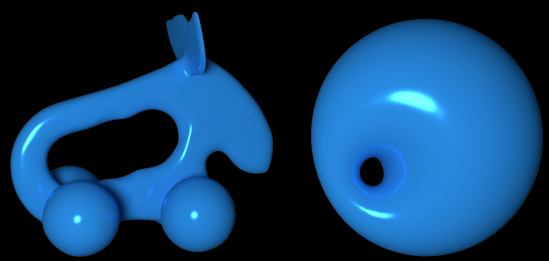
$\Rightarrow$ Robust and efficient
numerical integration

Outlook

Isogeometric
Subdivision Method



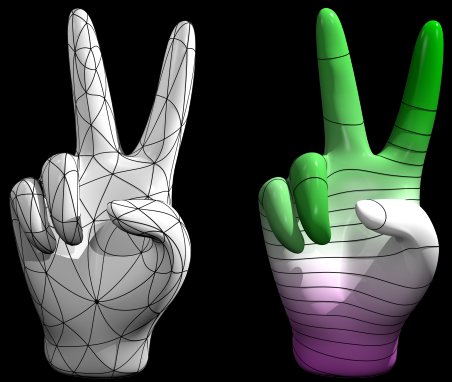
Parametric
Gradient Flows



$\Rightarrow$ Robust and efficient
numerical integration

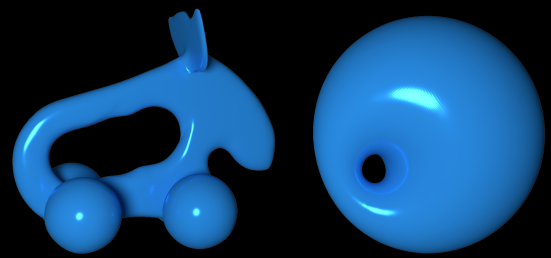
Outlook

Isogeometric
Subdivision Method



$\Rightarrow$ Robust and efficient
numerical integration

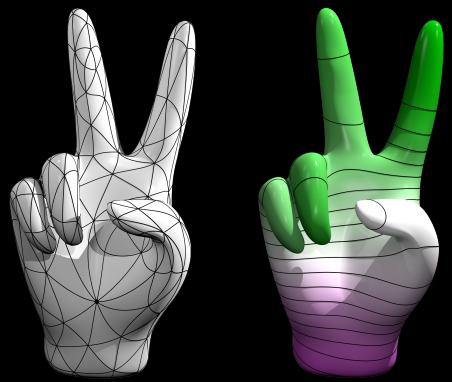
Parametric
Gradient Flows



$\Rightarrow$ Higher order
Time and Space
Discretization

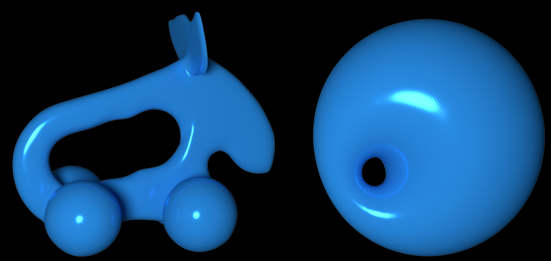
Outlook

Isogeometric
Subdivision Method



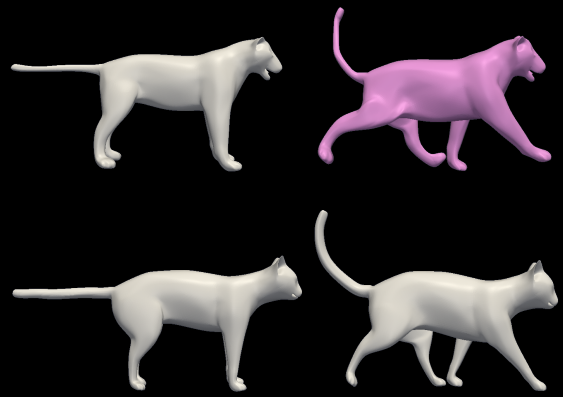
$\Rightarrow$ Robust and efficient
numerical integration

Parametric
Gradient Flows



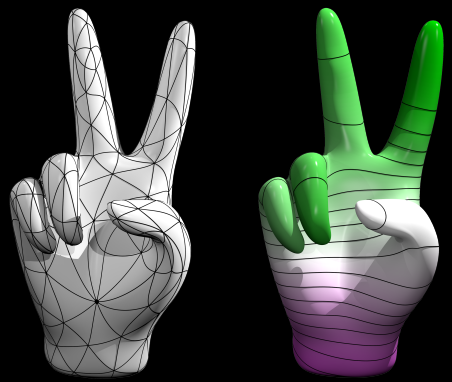
$\Rightarrow$ Higher order
Time and Space
Discretization

Shape Space



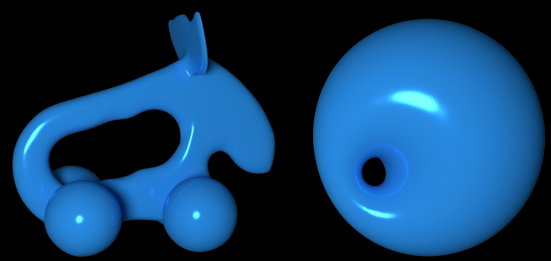
Outlook

Isogeometric
Subdivision Method



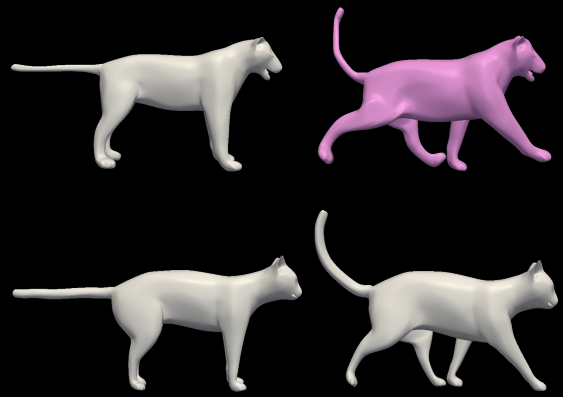
$\Rightarrow$ Robust and efficient
numerical integration

Parametric
Gradient Flows



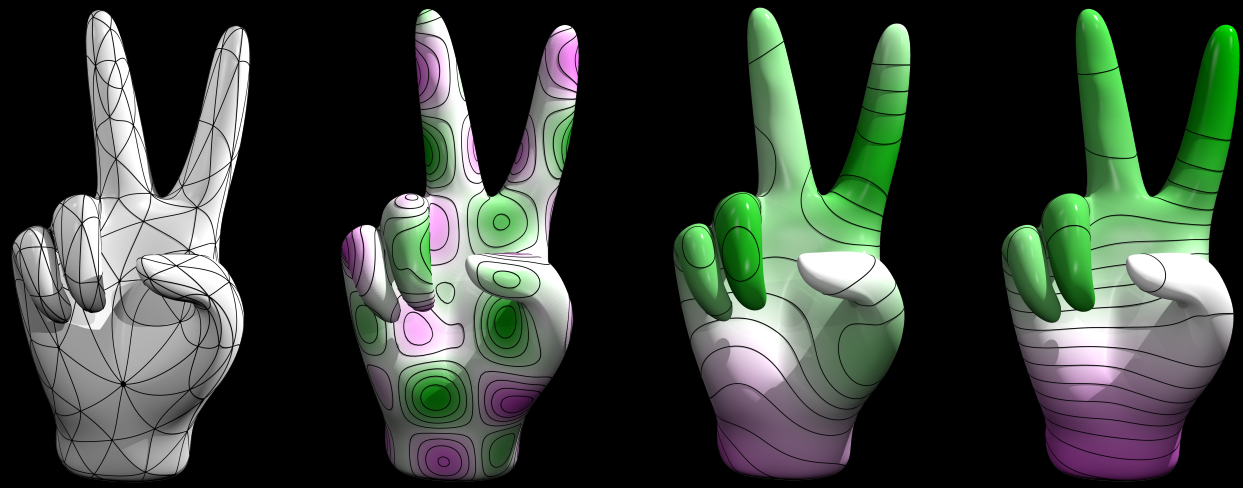
$\Rightarrow$ Higher order
Time and Space
Discretization

Shape Space



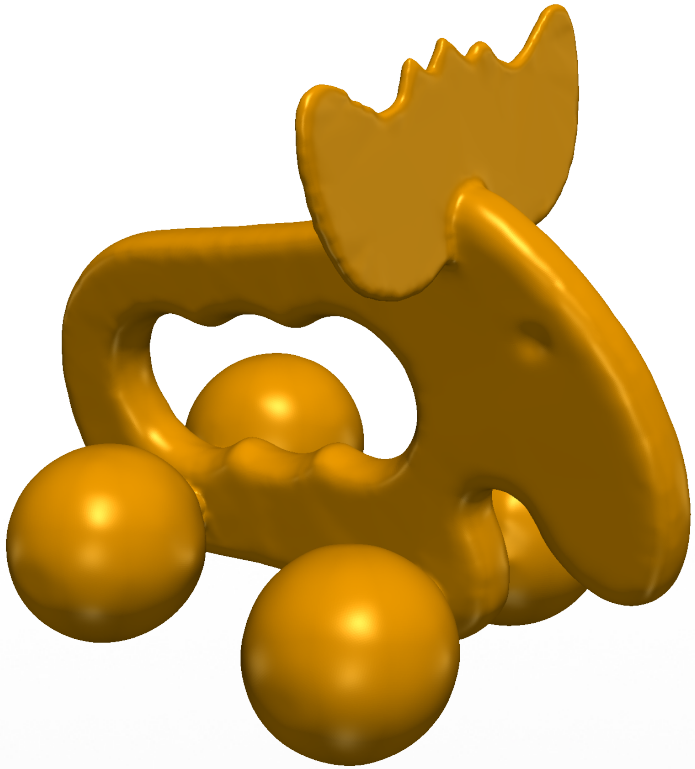
$\Rightarrow$ Bézier curves
and splines

Isogeometric Subdivision Method

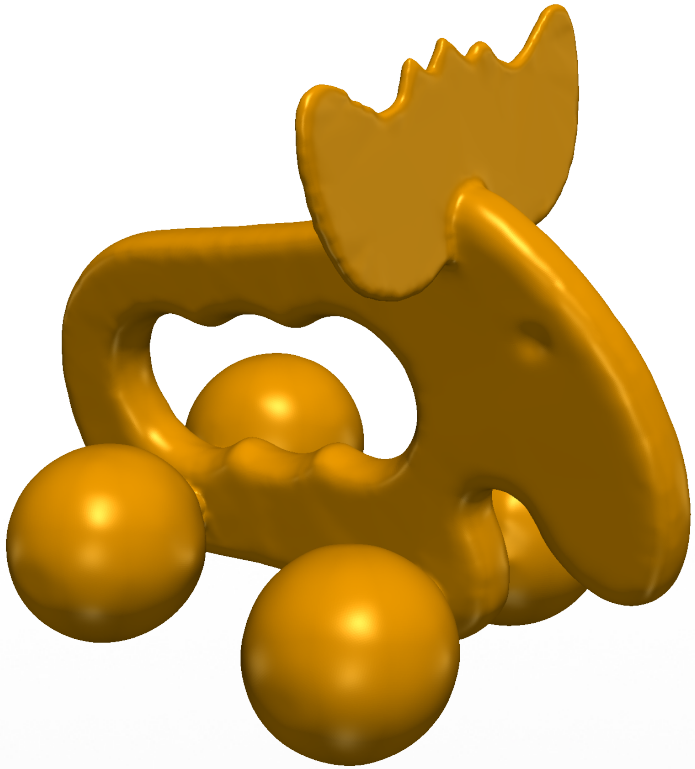


$\mathcal{M}$ f $-\Delta_{\mathcal{M}} u = f$ $\Delta^2_{\mathcal{M}} u = f$

Discretization Spaces

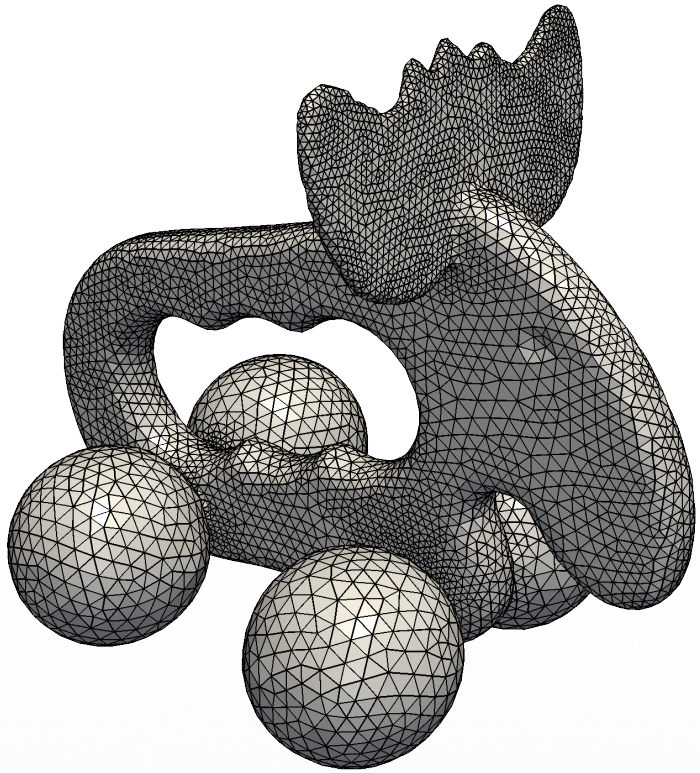


Discretization Spaces



How to discretize smooth surfaces?

Discretization Spaces



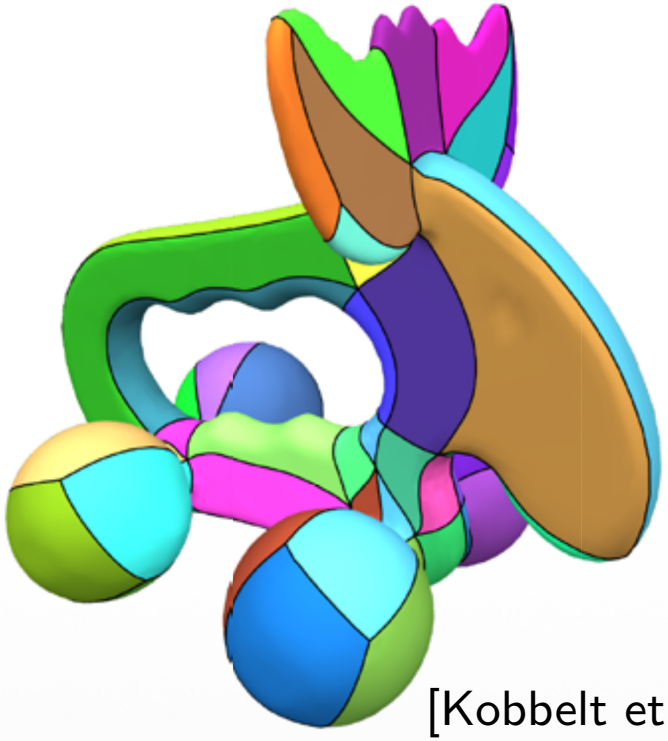
How to discretize smooth surfaces?

Classical FEM

✓ Flexibility

✗ Smoothness

Discretization Spaces



[Kobbelt et. al.'13]

How to discretize smooth surfaces?

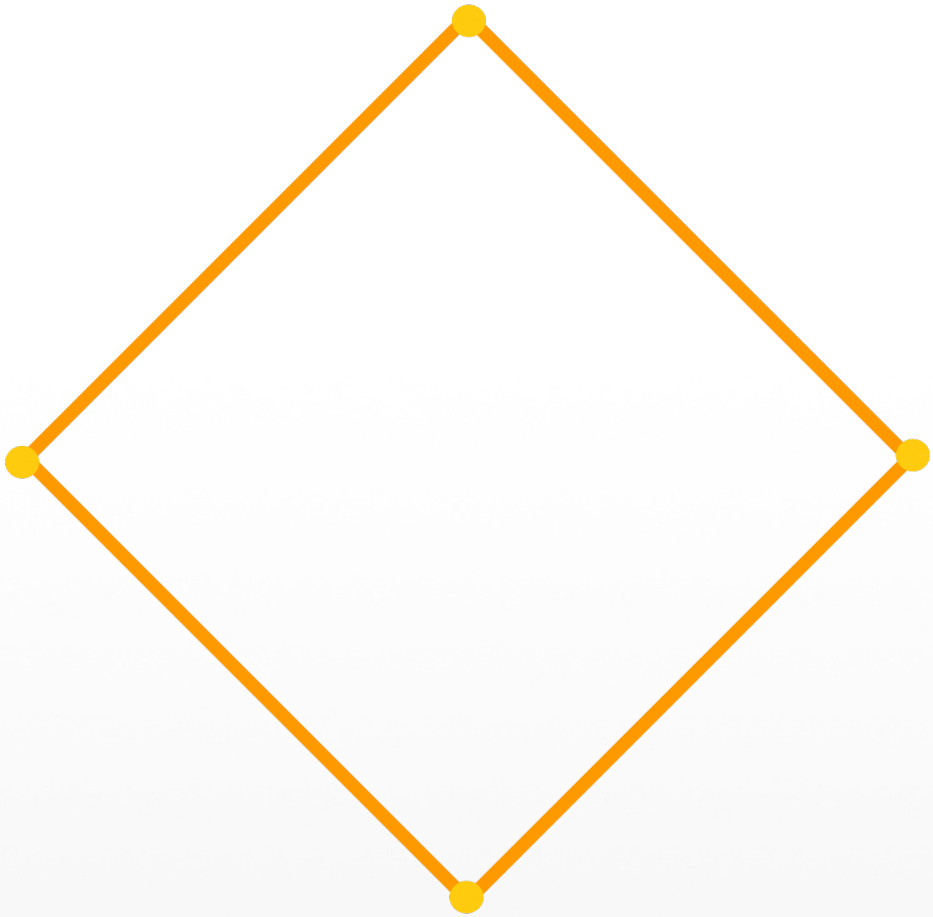
Classical FEM

- ✓ Flexibility
- ✗ Smoothness

NURBS based IgA

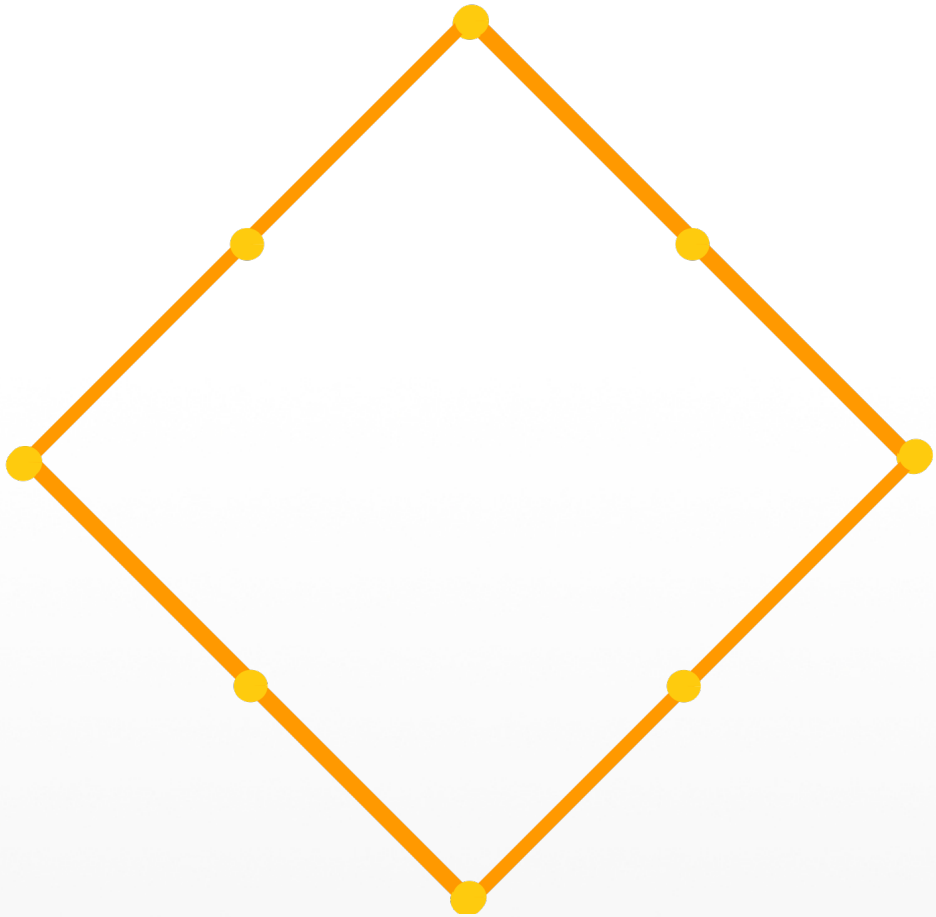
- ✗ Flexibility
- ✓ Smoothness

1D Subdivision



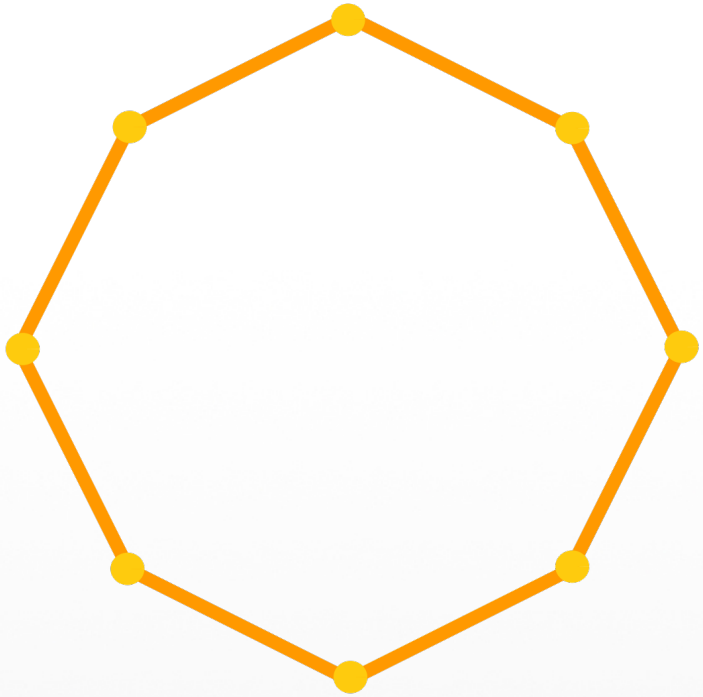
1D Subdivision

1. Subdivision Step
   $\rightsquigarrow$ split



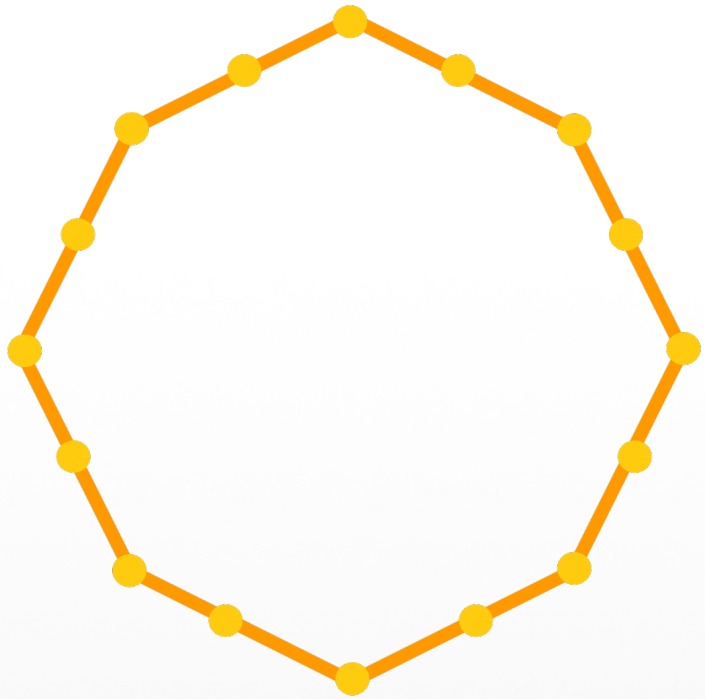
1D Subdivision

1. Subdivision Step
   - ⇝ split
   - ⇝ smooth



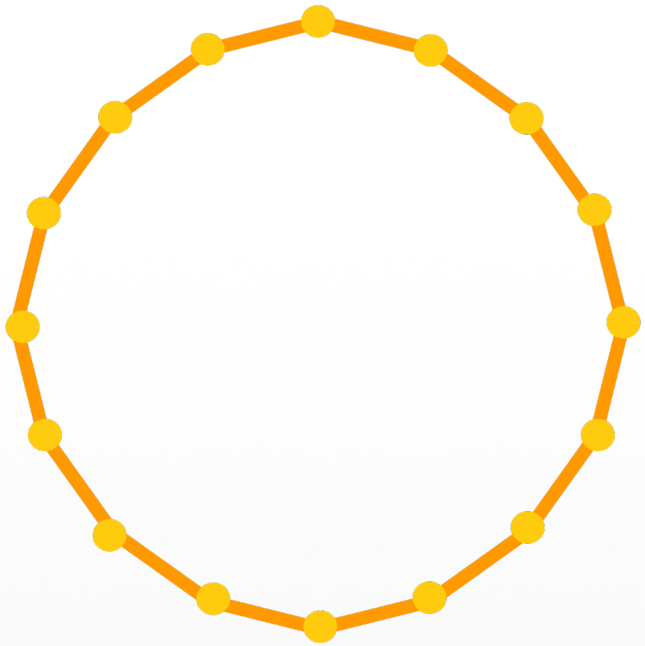
1D Subdivision

1. Subdivision Step
   - ⇝ split
   - ⇝ smooth
2. Subdivision Step
   - ⇝ split



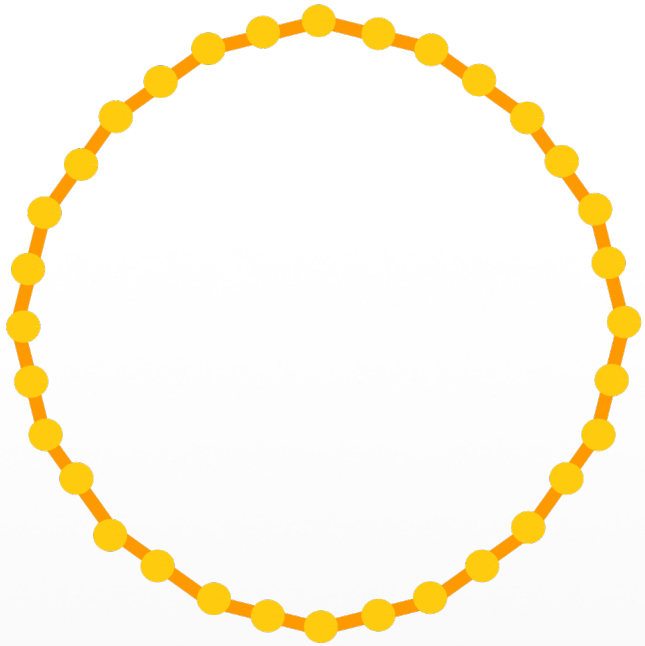
1D Subdivision

1. Subdivision Step
   - ⇝ split
   - ⇝ smooth
2. Subdivision Step
   - ⇝ split
   - ⇝ smooth



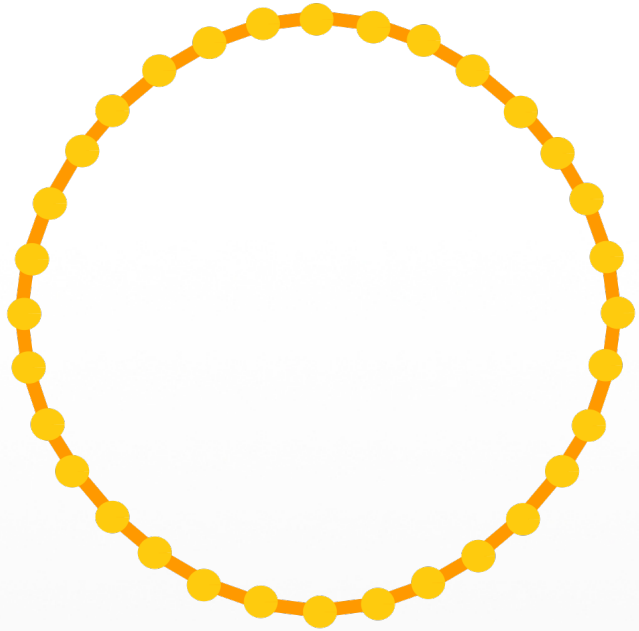
1D Subdivision

1. Subdivision Step
   - ⇝ split
   - ⇝ smooth
2. Subdivision Step
   - ⇝ split
   - ⇝ smooth
3. Subdivision Step
   - ⇝ split



1D Subdivision

1. Subdivision Step
   - ⇝ split
   - ⇝ smooth
2. Subdivision Step
   - ⇝ split
   - ⇝ smooth
3. Subdivision Step
   - ⇝ split
   - ⇝ smooth

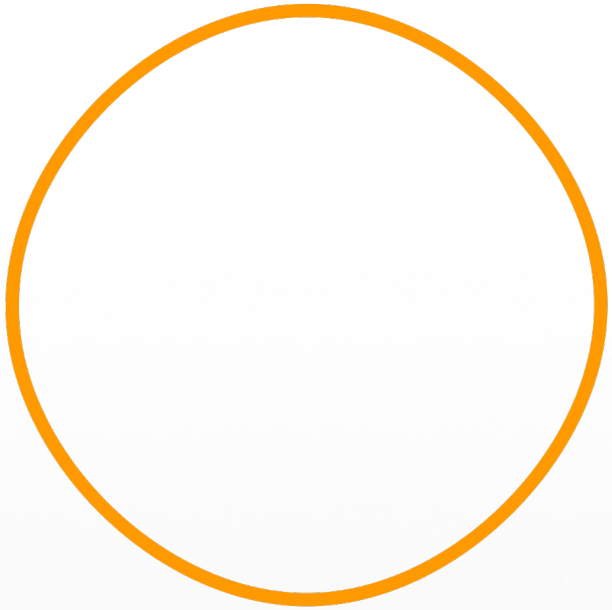


1D Subdivision

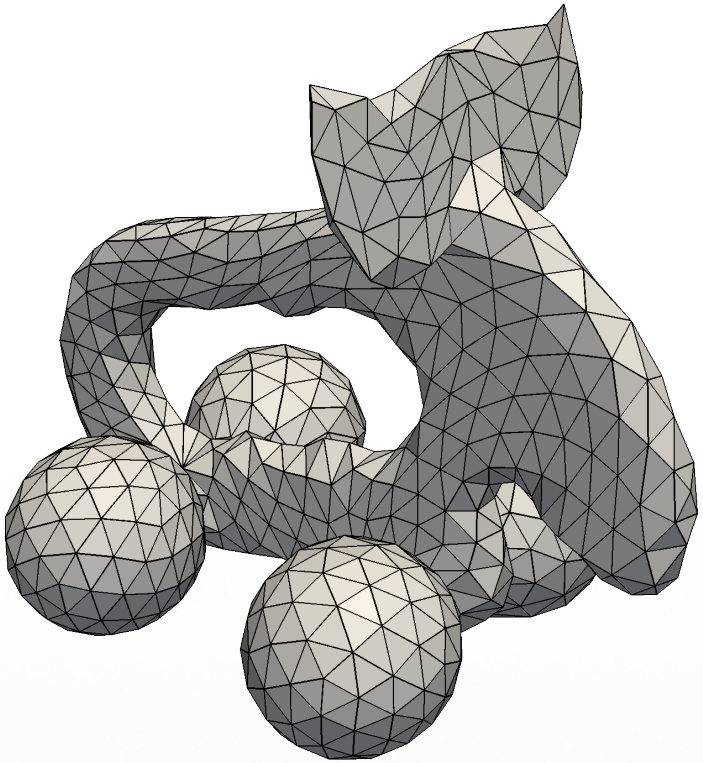
1. Subdivision Step
   - ⇝ split
   - ⇝ smooth
2. Subdivision Step
   - ⇝ split
   - ⇝ smooth
3. Subdivision Step
   - ⇝ split
   - ⇝ smooth

⋮

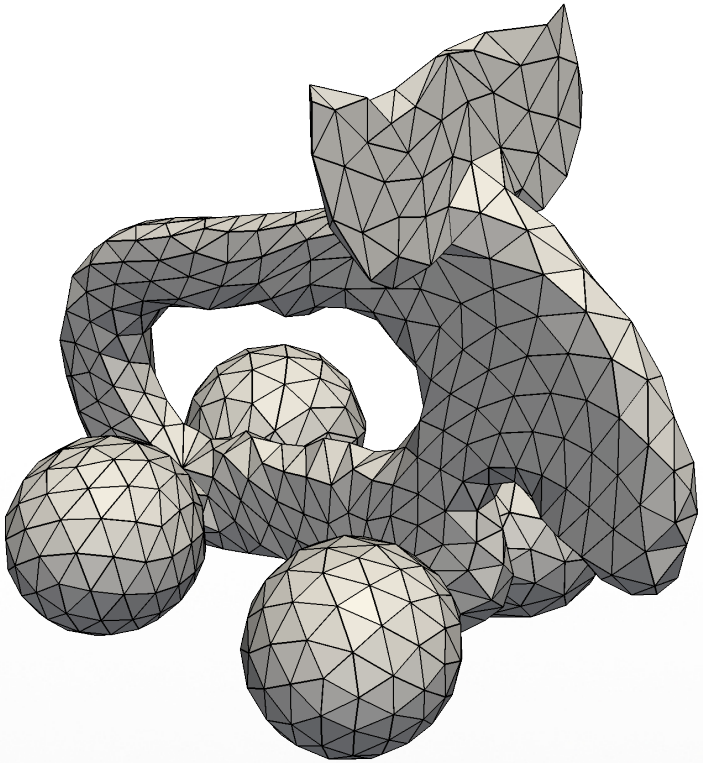
Limit curve

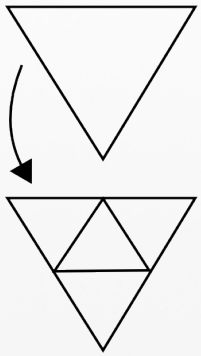


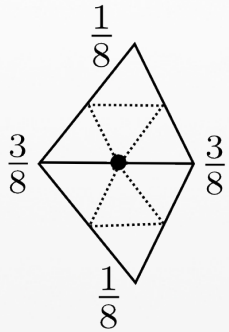
Subdivision Surfaces

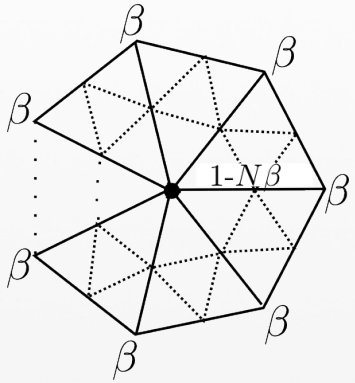


Subdivision Surfaces

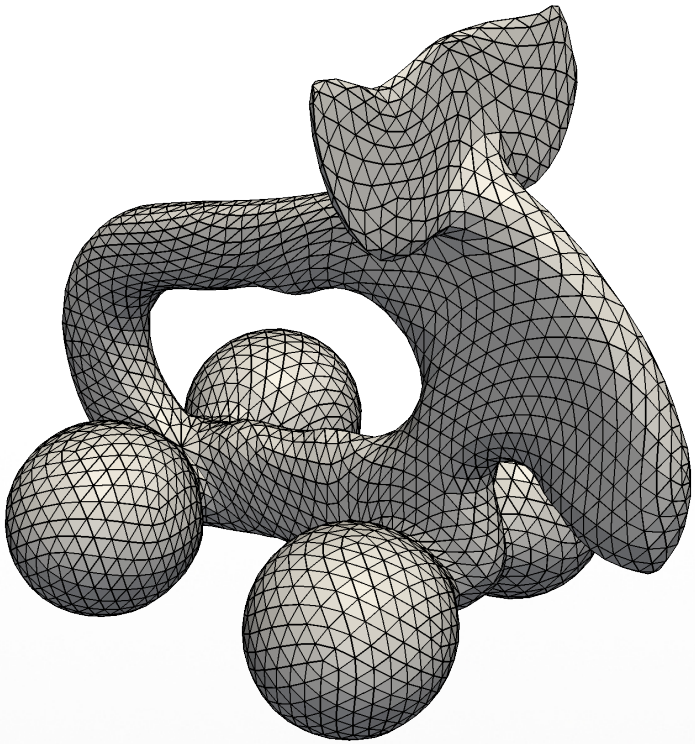


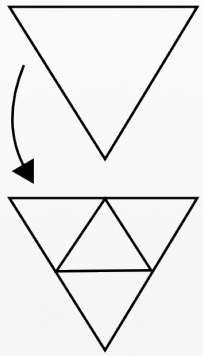


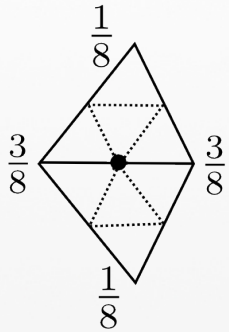


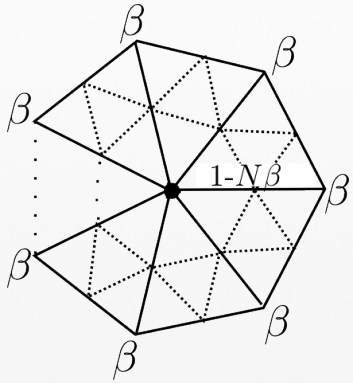


Subdivision Surfaces

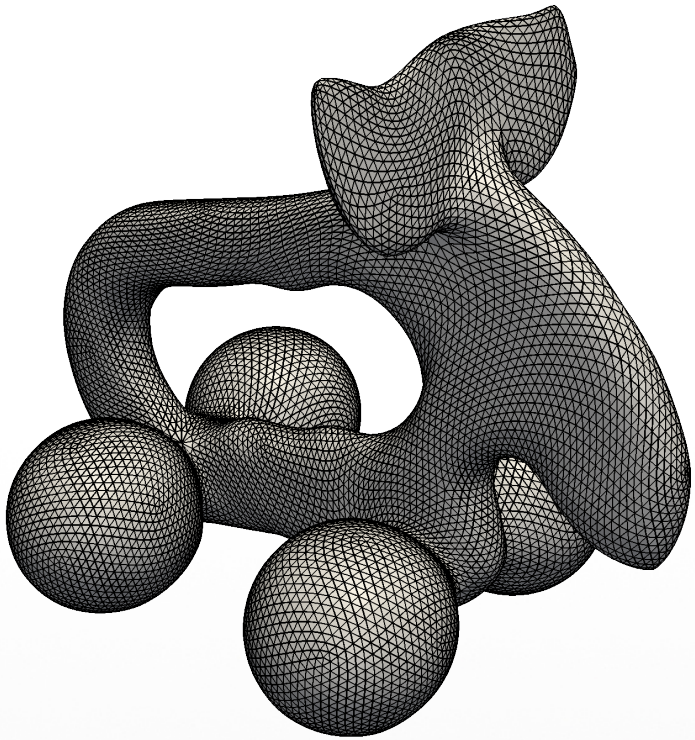


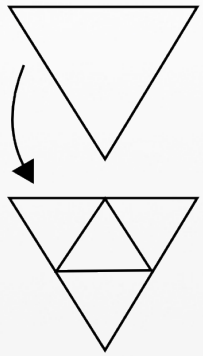


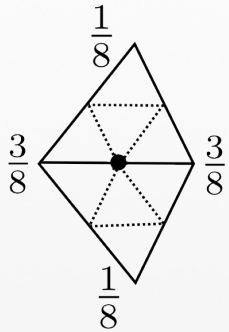


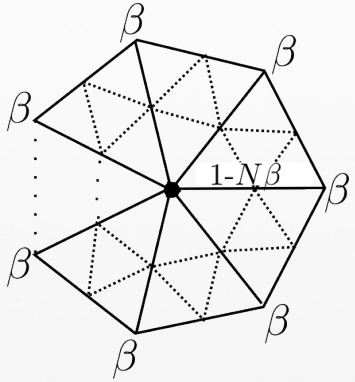


Subdivision Surfaces

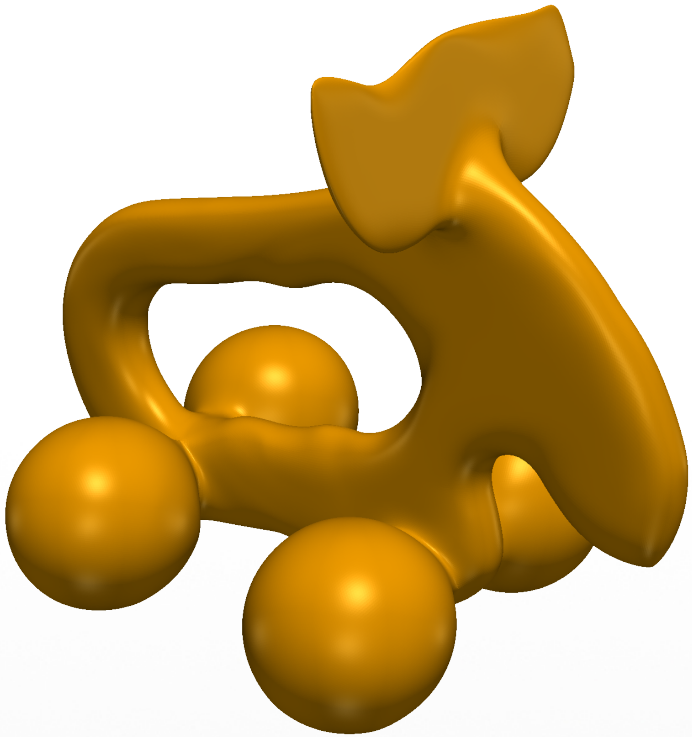


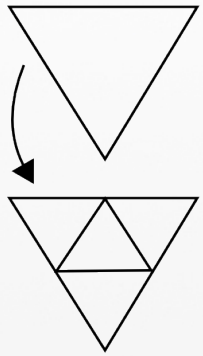


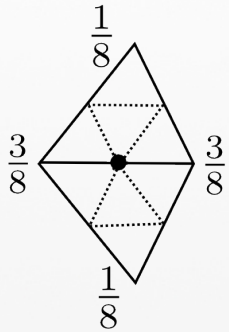


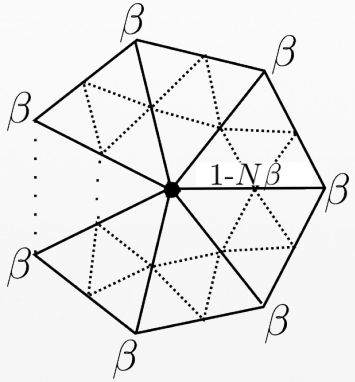


Subdivision Surfaces



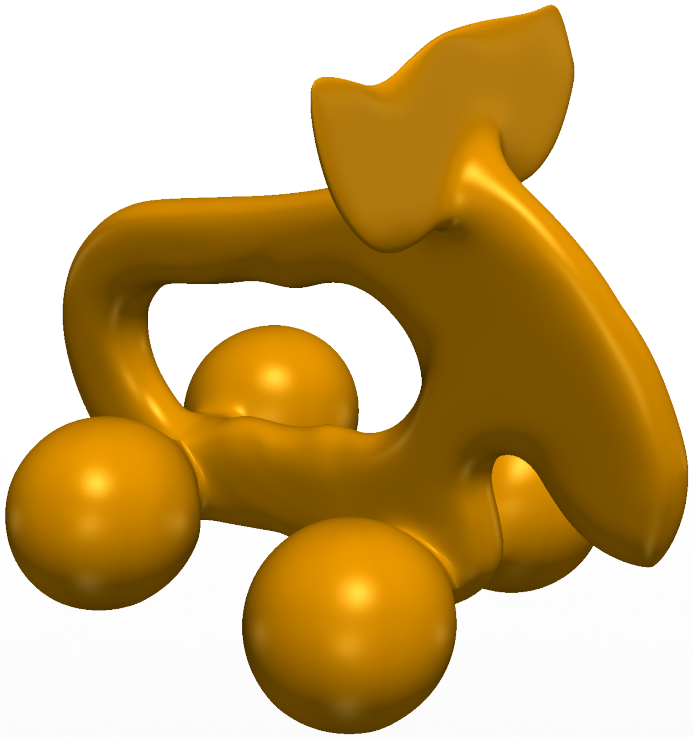


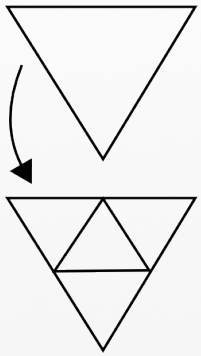


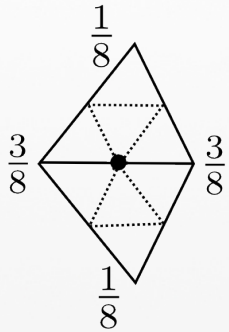


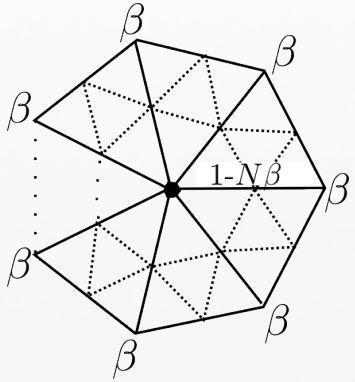
Subdivision Surfaces

- ✓ Flexibility
- ✓ Smoothness $\Rightarrow C^1$









Isogeometric Subdivision Method

- [Cirak, Ortiz and Schöder, 2000] proposed to use subdivision surfaces in finite element simulations for thin-shell structures

Isogeometric Subdivision Method

- [Cirak, Ortiz and Schöder, 2000] proposed to use subdivision surfaces in finite element simulations for thin-shell structures
- Isogeometric Analysis (IgA) was invented by [Hughes, Cottrell and Bazilevs, 2005] to unify CAD and FEM

Isogeometric Subdivision Method

- [Cirak, Ortiz and Schöder, 2000] proposed to use subdivision surfaces in finite element simulations for thin-shell structures
- Isogeometric Analysis (IgA) was invented by [Hughes, Cottrell and Bazilevs, 2005] to unify CAD and FEM
- NURBS are dominating CAD technology, but subdivision is nowadays used by CAD systems (e.g. PTC Creo, Solid Works)

Isogeometric Subdivision Method

- [Cirak, Ortiz and Schöder, 2000] proposed to use subdivision surfaces in finite element simulations for thin-shell structures
- Isogeometric Analysis (IgA) was invented by [Hughes, Cottrell and Bazilevs, 2005] to unify CAD and FEM
- NURBS are dominating CAD technology, but subdivision is nowadays used by CAD systems (e.g. PTC Creo, Solid Works)
- Numerical integration is a bottleneck of IgA: Based on higher order polynomials $\Rightarrow$ exact Gaussian quadrature is expensive

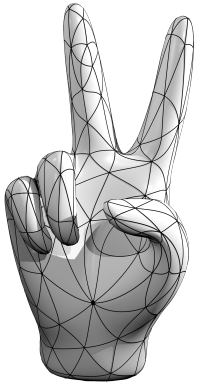
Isogeometric Subdivision Method

- [Cirak, Ortiz and Schöder, 2000] proposed to use subdivision surfaces in finite element simulations for thin-shell structures
- Isogeometric Analysis (IgA) was invented by [Hughes, Cottrell and Bazilevs, 2005] to unify CAD and FEM
- NURBS are dominating CAD technology, but subdivision is nowadays used by CAD systems (e.g. PTC Creo, Solid Works)
- Numerical integration is a bottleneck of IgA: Based on higher order polynomials $\Rightarrow$ exact Gaussian quadrature is expensive
- How to overcome expensive evaluation technique? $\rightarrow$ Reduced quadrature

Isogeometric Subdivision Method

- [Cirak, Ortiz and Schöder, 2000] proposed to use subdivision surfaces in finite element simulations for thin-shell structures
- Isogeometric Analysis (IgA) was invented by [Hughes, Cottrell and Bazilevs, 2005] to unify CAD and FEM
- NURBS are dominating CAD technology, but subdivision is nowadays used by CAD systems (e.g. PTC Creo, Solid Works)
- Numerical integration is a bottleneck of IgA: Based on higher order polynomials $\Rightarrow$ exact Gaussian quadrature is expensive
- How to overcome expensive evaluation technique? $\rightarrow$ Reduced quadrature
- [Cirak, Ortiz and Schöder, 2000] use barycenter rule (one point per element)

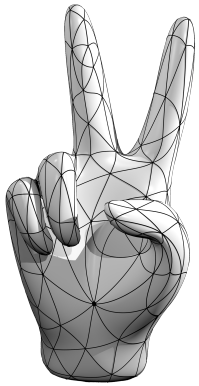
Reduced Quadrature



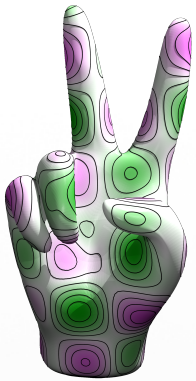
$\mathcal{M}$

- Given subdivision surface $\mathcal{M}$

Reduced Quadrature



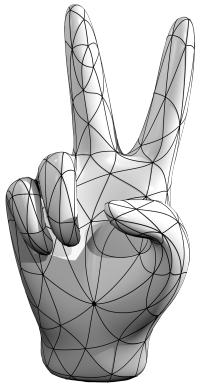
$\mathcal{M}$



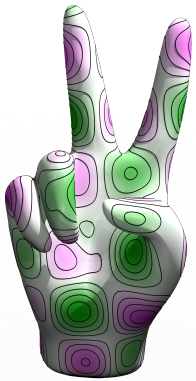
f

- Given subdivision surface $\mathcal{M}$
- Given right-hand side f

Reduced Quadrature



$\mathcal{M}$

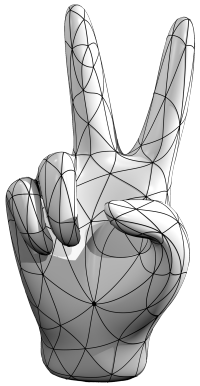


f

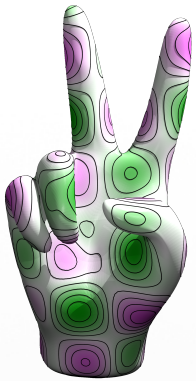
$$(\Delta_{\mathcal{M}})^2 u = f$$

- Given subdivision surface $\mathcal{M}$
- Given right-hand side f
- Solve bi-Laplacian on $\mathcal{M}$

Reduced Quadrature

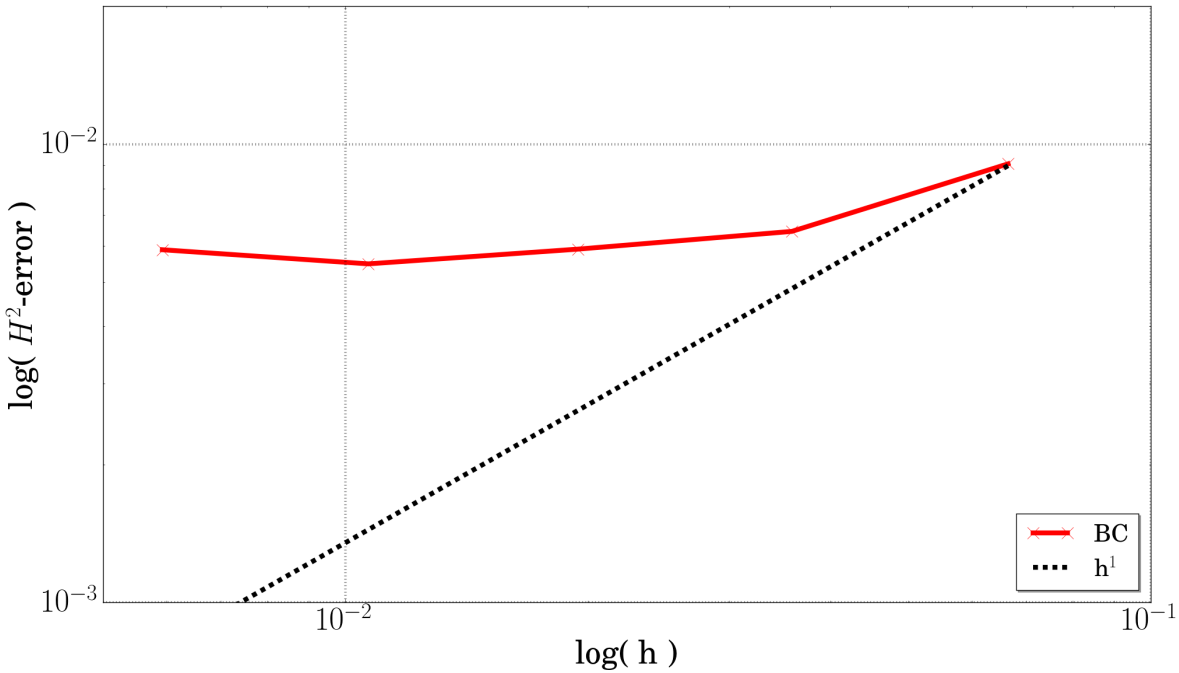


$\mathcal{M}$



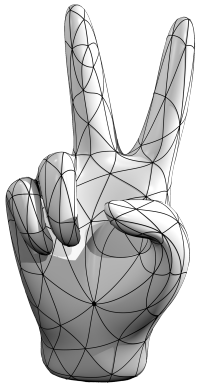
f

$$(\Delta_{\mathcal{M}})^2 u = f$$

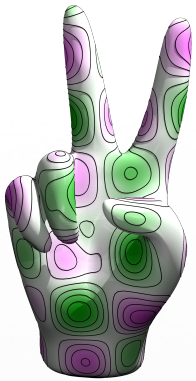


- Given subdivision surface $\mathcal{M}$
- Given right-hand side f
- Solve bi-Laplacian on $\mathcal{M}$
- Barycenter (BC) rule can fail!

Reduced Quadrature

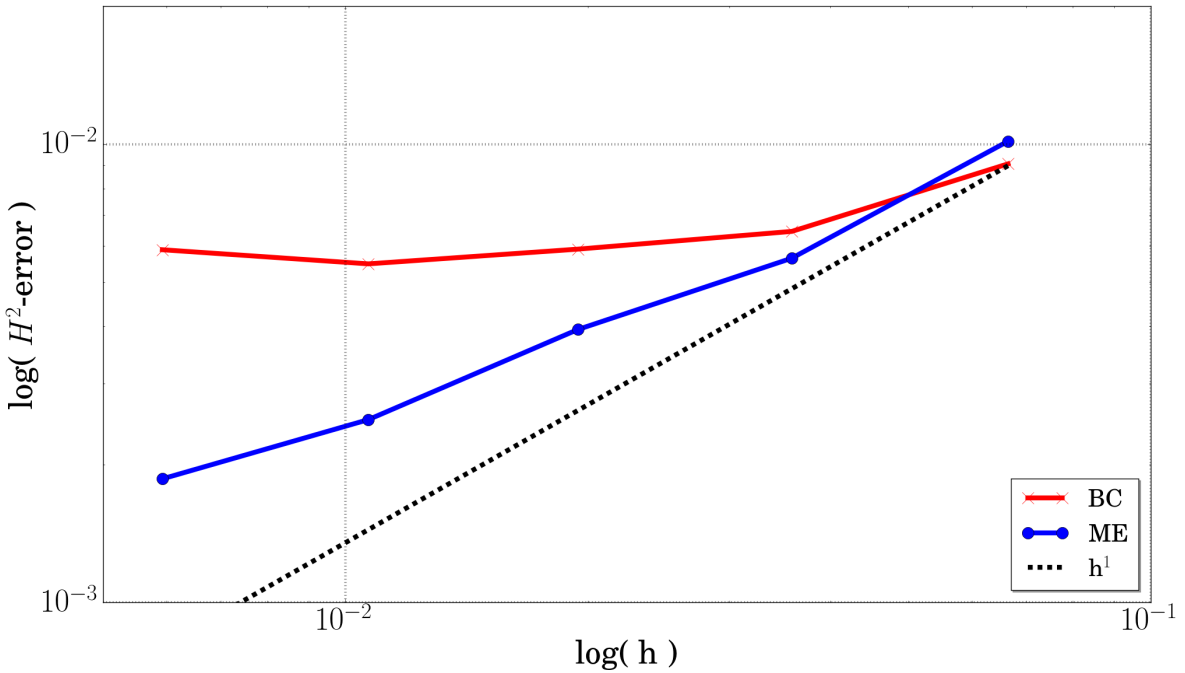


$\mathcal{M}$



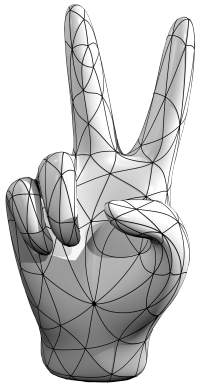
f

$$(\Delta_{\mathcal{M}})^2 u = f$$

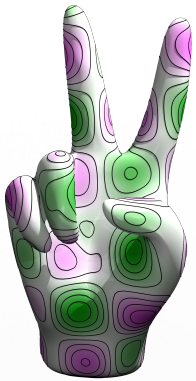


- Given subdivision surface $\mathcal{M}$
- Given right-hand side f
- Solve bi-Laplacian on $\mathcal{M}$
- Barycenter (BC) rule can fail!
- Mid-edge (ME) rule works fine, BUT is less efficient.

Reduced Quadrature

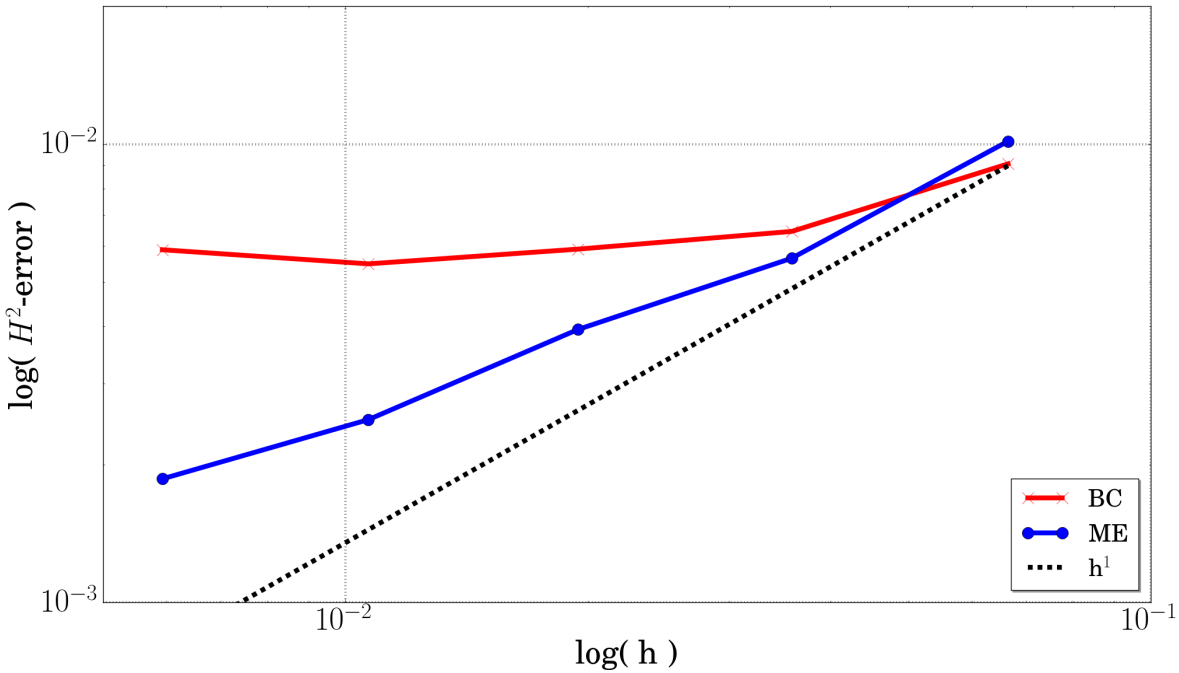


$\mathcal{M}$



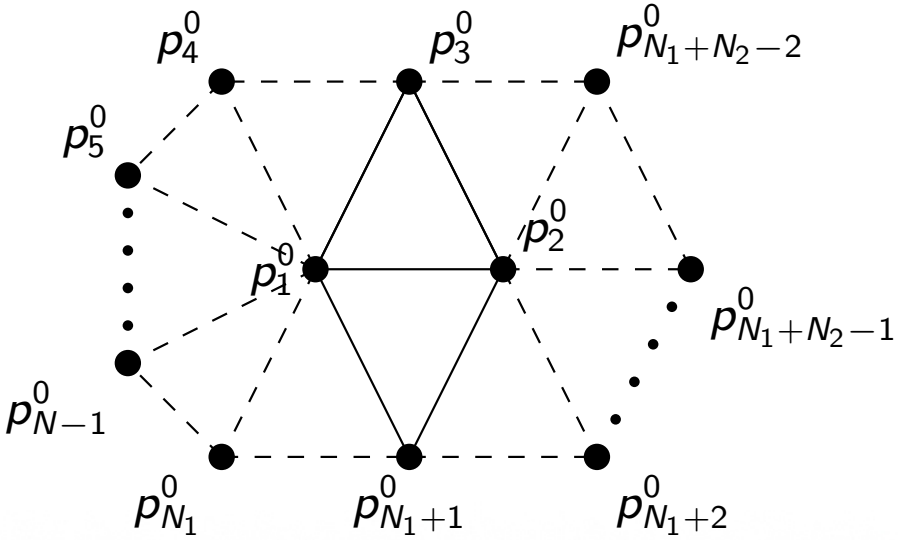
f

$$(\Delta_{\mathcal{M}})^2 u = f$$



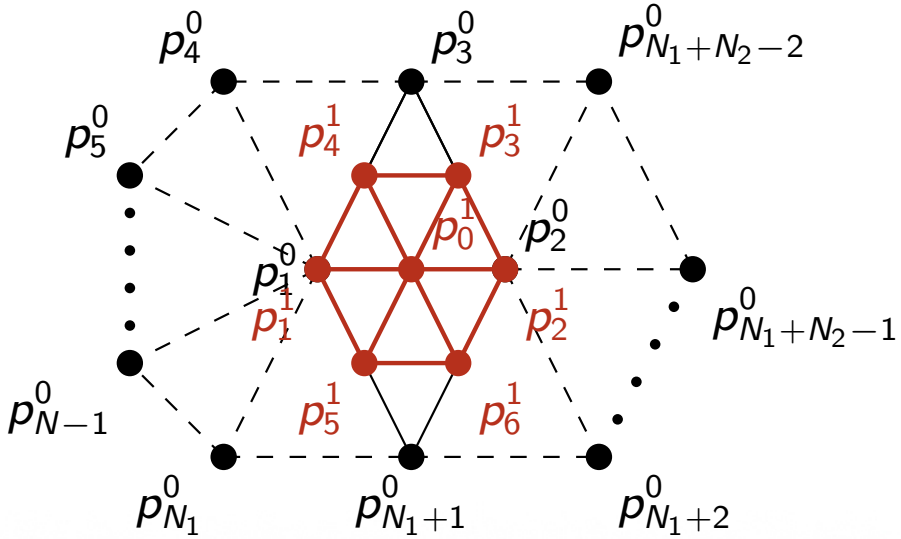
- Given subdivision surface $\mathcal{M}$
- Given right-hand side f
- Solve bi-Laplacian on $\mathcal{M}$
- Barycenter (BC) rule can fail!
- Mid-edge (ME) rule works fine, BUT is less efficient.
- Implementation of evaluation process is complex and time consuming.

Mid-edge Lookup Tables



$\mathbf{p}^0$

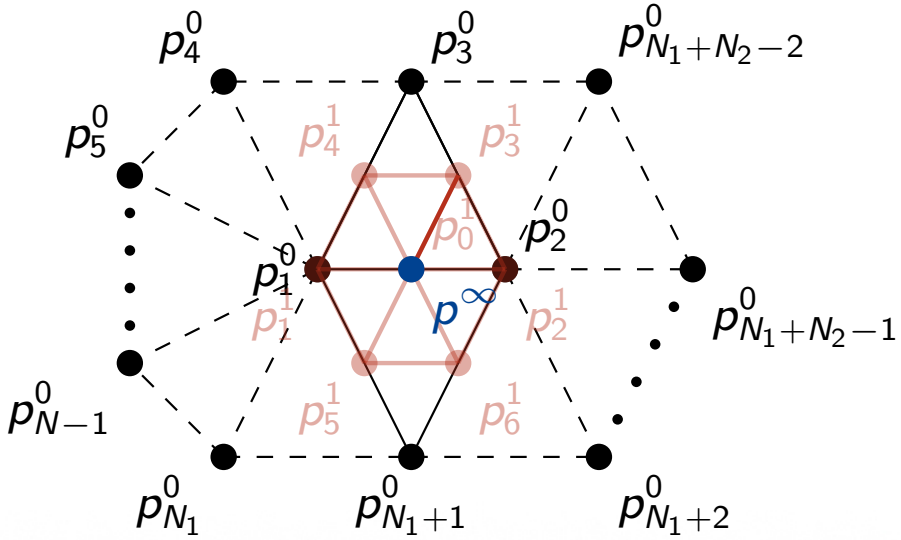
Mid-edge Lookup Tables



$$\mathbf{p}^1 = S\mathbf{p}^0$$

$$S = \begin{pmatrix}
\frac{3}{8} & \frac{3}{8} & \frac{1}{8} & 0 & 0 & \dots & 0 & 0 & \frac{1}{8} & 0 & 0 & \dots & 0 & 0 \\
\beta_2 & 1\text{-}\beta_2 \cdot N_2 & \beta_2 & 0 & 0 & \dots & 0 & 0 & \beta_2 & \beta_2 & \beta_2 & \dots & \beta_2 & \beta_2 \\
\frac{1}{8} & \frac{3}{8} & \frac{3}{8} & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & \dots & 0 & \frac{1}{8} \\
\frac{3}{8} & \frac{1}{8} & \frac{3}{8} & \frac{1}{8} & 0 & \dots & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 \\
1\text{-}\beta_1 \cdot N_1 & \beta_1 & \beta_1 & \beta_1 & \beta_1 & \dots & \beta_1 & \beta_1 & \beta_1 & 0 & 0 & \dots & 0 & 0 \\
\frac{3}{8} & \frac{1}{8} & 0 & 0 & 0 & \dots & 0 & \frac{1}{8} & \frac{3}{8} & 0 & 0 & \dots & 0 & 0 \\
\frac{1}{8} & \frac{3}{8} & 0 & 0 & 0 & \dots & 0 & 0 & \frac{3}{8} & \frac{1}{8} & 0 & \dots & 0 & 0
\end{pmatrix}$$

Mid-edge Lookup Tables

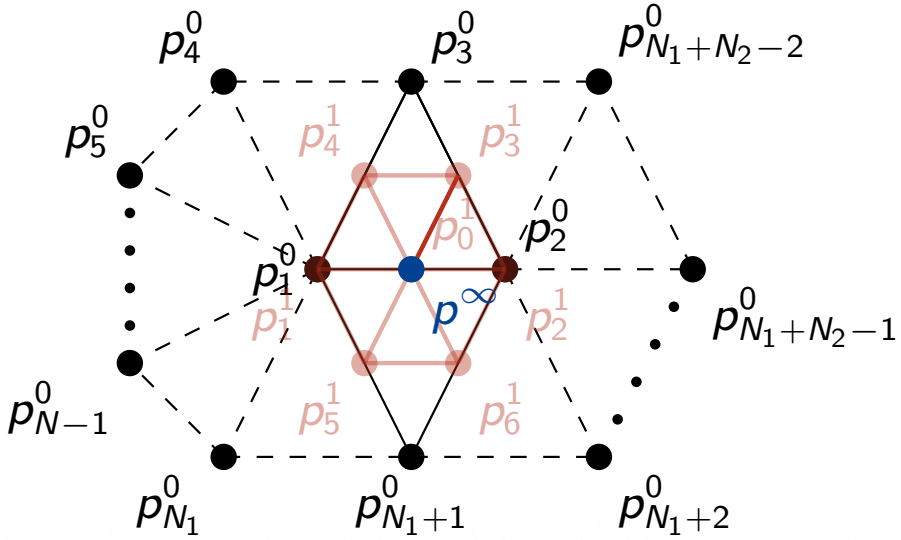


$$\mathbf{p}^\infty = L\mathbf{p}^1 = L \cdot S\mathbf{p}^0$$

$$S = \begin{pmatrix}
\frac{3}{8} & \frac{3}{8} & \frac{1}{8} & 0 & 0 & \dots & 0 & 0 & \frac{1}{8} & 0 & 0 & \dots & 0 & 0 \\
\beta_2 & 1\text{-}\beta_2 \cdot N_2 & \beta_2 & 0 & 0 & \dots & 0 & 0 & \beta_2 & \beta_2 & \beta_2 & \dots & \beta_2 & \beta_2 \\
\frac{1}{8} & \frac{3}{8} & \frac{3}{8} & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & \dots & 0 & \frac{1}{8} \\
\frac{3}{8} & \frac{1}{8} & \frac{3}{8} & \frac{1}{8} & 0 & \dots & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 \\
1\text{-}\beta_1 \cdot N_1 & \beta_1 & \beta_1 & \beta_1 & \beta_1 & \dots & \beta_1 & \beta_1 & \beta_1 & 0 & 0 & \dots & 0 & 0 \\
\frac{3}{8} & \frac{1}{8} & 0 & 0 & 0 & \dots & 0 & \frac{1}{8} & \frac{3}{8} & 0 & 0 & \dots & 0 & 0 \\
\frac{1}{8} & \frac{3}{8} & 0 & 0 & 0 & \dots & 0 & 0 & \frac{3}{8} & \frac{1}{8} & 0 & \dots & 0 & 0
\end{pmatrix}$$

$$L = \begin{pmatrix}
\frac{1}{2} & \frac{1}{12} & \frac{1}{12} & \frac{1}{12} & \frac{1}{12} & \frac{1}{12} & \frac{1}{12} \\
0 & \frac{2}{3} & \frac{1}{3} & -\frac{1}{3} & -\frac{2}{3} & -\frac{1}{3} & \frac{1}{3} \\
0 & \frac{1}{3} & \frac{2}{3} & \frac{1}{3} & -\frac{1}{3} & -\frac{2}{3} & -\frac{1}{3} \\
-8 & 4 & 0 & 0 & 4 & 0 & 0 \\
-4 & 2 & 2 & -2 & 2 & 2 & -2 \\
-8 & 0 & 4 & 0 & 0 & 4 & 0
\end{pmatrix}$$

Mid-edge Lookup Tables



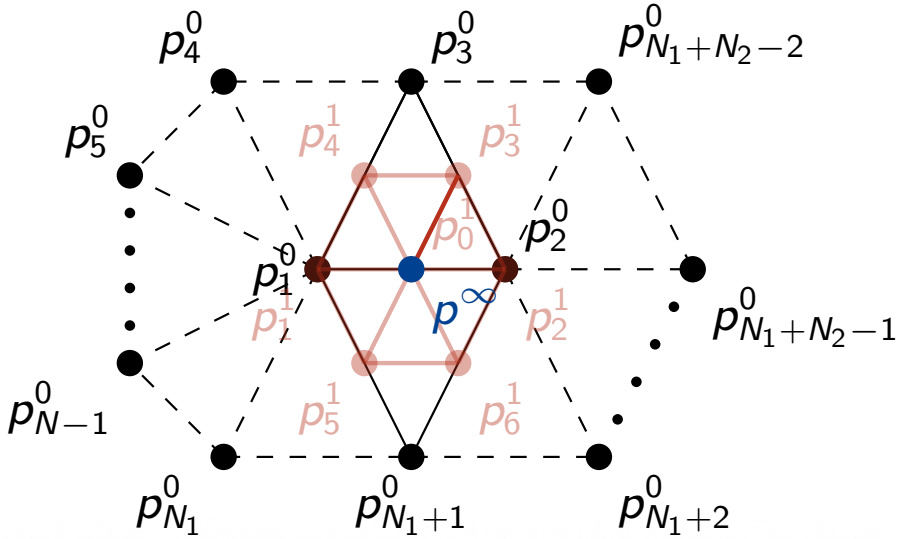
$$\mathbf{p}^\infty = L\mathbf{p}^1 = L \cdot S\mathbf{p}^0$$

- Position, first and second derivatives

$$S = \begin{pmatrix}
\frac{3}{8} & \frac{3}{8} & \frac{1}{8} & 0 & 0 & \dots & 0 & 0 & \frac{1}{8} & 0 & 0 & \dots & 0 & 0 \\
\beta_2 & \text{1-}\beta_2 \cdot N_2 & \beta_2 & 0 & 0 & \dots & 0 & 0 & \beta_2 & \beta_2 & \beta_2 & \dots & \beta_2 & \beta_2 \\
\frac{1}{8} & \frac{3}{8} & \frac{3}{8} & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & \dots & 0 & \frac{1}{8} \\
\frac{3}{8} & \frac{1}{8} & \frac{3}{8} & \frac{1}{8} & 0 & \dots & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 \\
\text{1-}\beta_1 \cdot N_1 & \beta_1 & \beta_1 & \beta_1 & \beta_1 & \dots & \beta_1 & \beta_1 & \beta_1 & 0 & 0 & \dots & 0 & 0 \\
\frac{3}{8} & \frac{1}{8} & 0 & 0 & 0 & \dots & 0 & \frac{1}{8} & \frac{3}{8} & 0 & 0 & \dots & 0 & 0 \\
\frac{1}{8} & \frac{3}{8} & 0 & 0 & 0 & \dots & 0 & 0 & \frac{3}{8} & \frac{1}{8} & 0 & \dots & 0 & 0
\end{pmatrix}$$

$$L = \begin{pmatrix}
\frac{1}{2} & \frac{1}{12} & \frac{1}{12} & \frac{1}{12} & \frac{1}{12} & \frac{1}{12} & \frac{1}{12} \\
0 & \frac{2}{3} & \frac{1}{3} & -\frac{1}{3} & -\frac{2}{3} & -\frac{1}{3} & \frac{1}{3} \\
0 & \frac{1}{3} & \frac{2}{3} & \frac{1}{3} & -\frac{1}{3} & -\frac{2}{3} & -\frac{1}{3} \\
-8 & 4 & 0 & 0 & 4 & 0 & 0 \\
-4 & 2 & 2 & -2 & 2 & 2 & -2 \\
-8 & 0 & 4 & 0 & 0 & 4 & 0
\end{pmatrix}$$

Mid-edge Lookup Tables



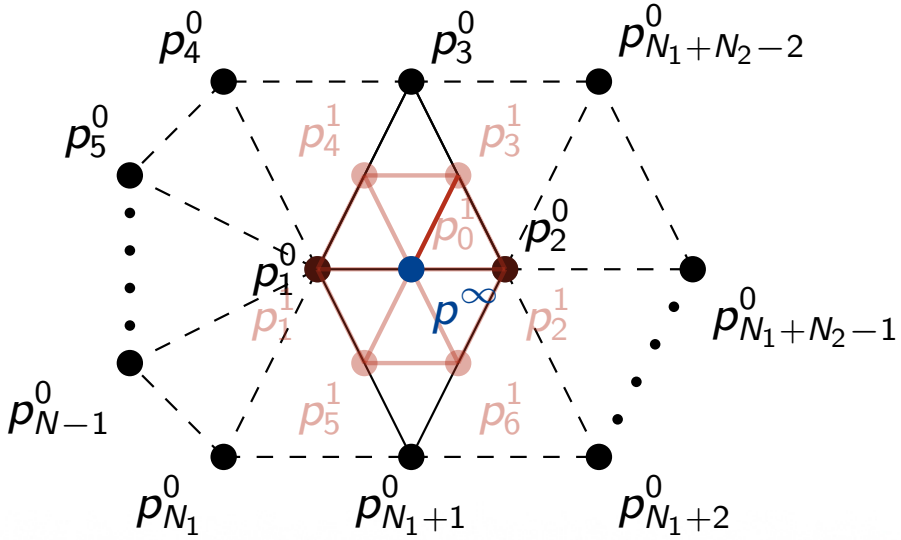
$$\mathbf{p}^\infty = L\mathbf{p}^1 = L \cdot S\mathbf{p}^0$$

- Position, first and second derivatives
- Easy to implement

$$S = \begin{pmatrix}
\frac{3}{8} & \frac{3}{8} & \frac{1}{8} & 0 & 0 & \dots & 0 & 0 & \frac{1}{8} & 0 & 0 & \dots & 0 & 0 \\
\beta_2 & 1\text{-}\beta_2 \cdot N_2 & \beta_2 & 0 & 0 & \dots & 0 & 0 & \beta_2 & \beta_2 & \beta_2 & \dots & \beta_2 & \beta_2 \\
\frac{1}{8} & \frac{3}{8} & \frac{3}{8} & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & \dots & 0 & \frac{1}{8} \\
\frac{3}{8} & \frac{1}{8} & \frac{3}{8} & \frac{1}{8} & 0 & \dots & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 \\
1\text{-}\beta_1 \cdot N_1 & \beta_1 & \beta_1 & \beta_1 & \beta_1 & \dots & \beta_1 & \beta_1 & \beta_1 & 0 & 0 & \dots & 0 & 0 \\
\frac{3}{8} & \frac{1}{8} & 0 & 0 & 0 & \dots & 0 & \frac{1}{8} & \frac{3}{8} & 0 & 0 & \dots & 0 & 0 \\
\frac{1}{8} & \frac{3}{8} & 0 & 0 & 0 & \dots & 0 & 0 & \frac{3}{8} & \frac{1}{8} & 0 & \dots & 0 & 0
\end{pmatrix}$$

$$L = \begin{pmatrix}
\frac{1}{2} & \frac{1}{12} & \frac{1}{12} & \frac{1}{12} & \frac{1}{12} & \frac{1}{12} & \frac{1}{12} \\
0 & \frac{2}{3} & \frac{1}{3} & -\frac{1}{3} & -\frac{2}{3} & -\frac{1}{3} & \frac{1}{3} \\
0 & \frac{1}{3} & \frac{2}{3} & \frac{1}{3} & -\frac{1}{3} & -\frac{2}{3} & -\frac{1}{3} \\
-8 & 4 & 0 & 0 & 4 & 0 & 0 \\
-4 & 2 & 2 & -2 & 2 & 2 & -2 \\
-8 & 0 & 4 & 0 & 0 & 4 & 0
\end{pmatrix}$$

Mid-edge Lookup Tables



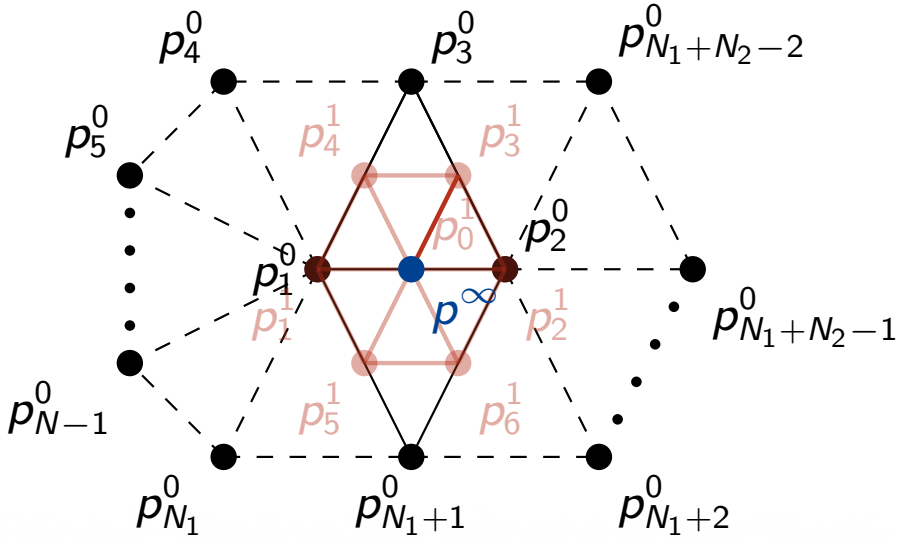
$$\mathbf{p}^\infty = L\mathbf{p}^1 = L \cdot S\mathbf{p}^0$$

- Position, first and second derivatives
- Easy to implement
- Faster than barycenter rule

$$S = \begin{pmatrix}
\frac{3}{8} & \frac{3}{8} & \frac{1}{8} & 0 & 0 & \dots & 0 & 0 & \frac{1}{8} & 0 & 0 & \dots & 0 & 0 \\
\beta_2 & 1\text{-}\beta_2 \cdot N_2 & \beta_2 & 0 & 0 & \dots & 0 & 0 & \beta_2 & \beta_2 & \beta_2 & \dots & \beta_2 & \beta_2 \\
\frac{1}{8} & \frac{3}{8} & \frac{3}{8} & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & \dots & 0 & \frac{1}{8} \\
\frac{3}{8} & \frac{1}{8} & \frac{3}{8} & \frac{1}{8} & 0 & \dots & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 \\
1\text{-}\beta_1 \cdot N_1 & \beta_1 & \beta_1 & \beta_1 & \beta_1 & \dots & \beta_1 & \beta_1 & \beta_1 & 0 & 0 & \dots & 0 & 0 \\
\frac{3}{8} & \frac{1}{8} & 0 & 0 & 0 & \dots & 0 & \frac{1}{8} & \frac{3}{8} & 0 & 0 & \dots & 0 & 0 \\
\frac{1}{8} & \frac{3}{8} & 0 & 0 & 0 & \dots & 0 & 0 & \frac{3}{8} & \frac{1}{8} & 0 & \dots & 0 & 0
\end{pmatrix}$$

$$L = \begin{pmatrix}
\frac{1}{2} & \frac{1}{12} & \frac{1}{12} & \frac{1}{12} & \frac{1}{12} & \frac{1}{12} & \frac{1}{12} \\
0 & \frac{2}{3} & \frac{1}{3} & -\frac{1}{3} & -\frac{2}{3} & -\frac{1}{3} & \frac{1}{3} \\
0 & \frac{1}{3} & \frac{2}{3} & \frac{1}{3} & -\frac{1}{3} & -\frac{2}{3} & -\frac{1}{3} \\
-8 & 4 & 0 & 0 & 4 & 0 & 0 \\
-4 & 2 & 2 & -2 & 2 & 2 & -2 \\
-8 & 0 & 4 & 0 & 0 & 4 & 0
\end{pmatrix}$$

Mid-edge Lookup Tables



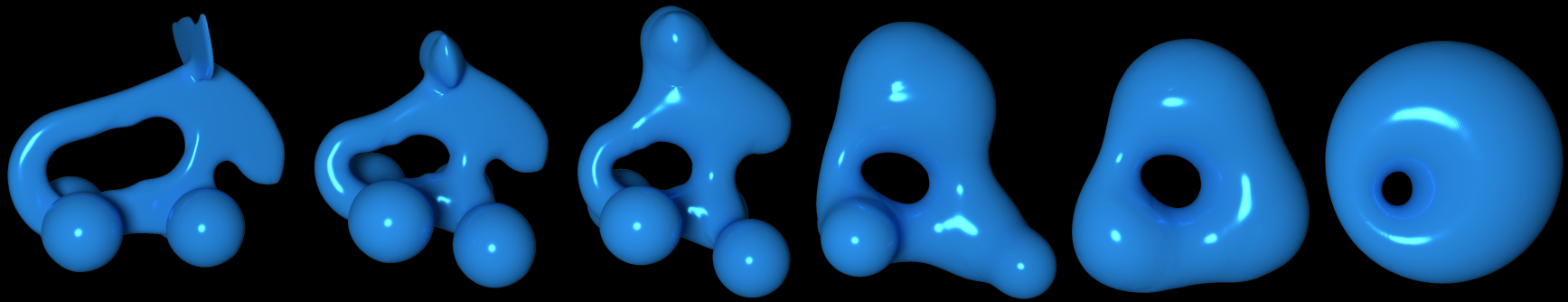
$$\mathbf{p}^\infty = L\mathbf{p}^1 = L \cdot S\mathbf{p}^0$$

- Position, first and second derivatives
- Easy to implement
- Faster than barycenter rule
- Arbitrary connectivity

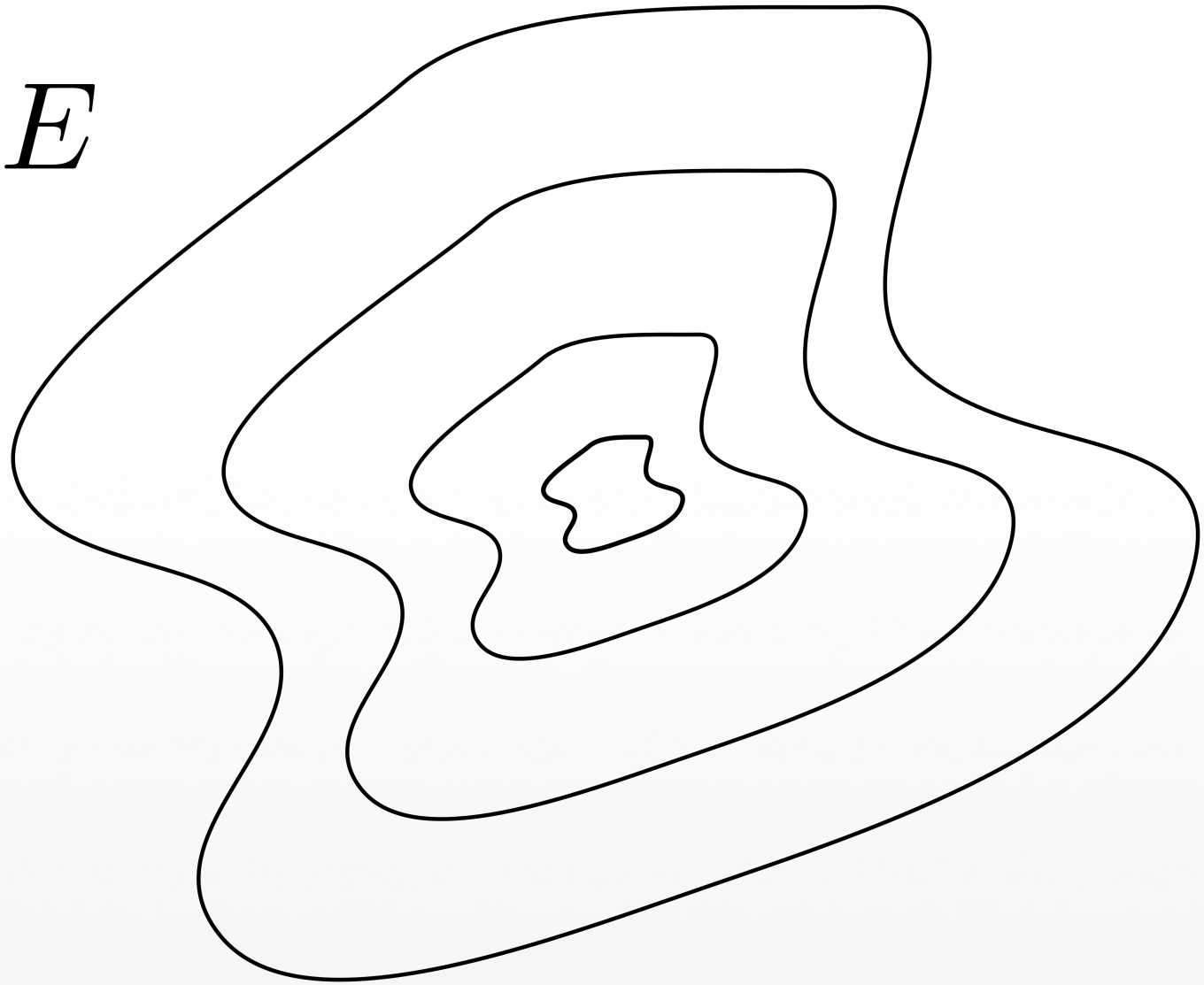
$$S = \begin{pmatrix}
\frac{3}{8} & \frac{3}{8} & \frac{1}{8} & 0 & 0 & \dots & 0 & 0 & \frac{1}{8} & 0 & 0 & \dots & 0 & 0 \\
\beta_2 & \text{1-}\beta_2 \cdot N_2 & \beta_2 & 0 & 0 & \dots & 0 & 0 & \beta_2 & \beta_2 & \beta_2 & \dots & \beta_2 & \beta_2 \\
\frac{1}{8} & \frac{3}{8} & \frac{3}{8} & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & \dots & 0 & \frac{1}{8} \\
\frac{3}{8} & \frac{1}{8} & \frac{3}{8} & \frac{1}{8} & 0 & \dots & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 \\
\text{1-}\beta_1 \cdot N_1 & \beta_1 & \beta_1 & \beta_1 & \beta_1 & \dots & \beta_1 & \beta_1 & \beta_1 & 0 & 0 & \dots & 0 & 0 \\
\frac{3}{8} & \frac{1}{8} & 0 & 0 & 0 & \dots & 0 & \frac{1}{8} & \frac{3}{8} & 0 & 0 & \dots & 0 & 0 \\
\frac{1}{8} & \frac{3}{8} & 0 & 0 & 0 & \dots & 0 & 0 & \frac{3}{8} & \frac{1}{8} & 0 & \dots & 0 & 0
\end{pmatrix}$$

$$L = \begin{pmatrix}
\frac{1}{2} & \frac{1}{12} & \frac{1}{12} & \frac{1}{12} & \frac{1}{12} & \frac{1}{12} & \frac{1}{12} \\
0 & \frac{2}{3} & \frac{1}{3} & -\frac{1}{3} & -\frac{2}{3} & -\frac{1}{3} & \frac{1}{3} \\
0 & \frac{1}{3} & \frac{2}{3} & \frac{1}{3} & -\frac{1}{3} & -\frac{2}{3} & -\frac{1}{3} \\
-8 & 4 & 0 & 0 & 4 & 0 & 0 \\
-4 & 2 & 2 & -2 & 2 & 2 & -2 \\
-8 & 0 & 4 & 0 & 0 & 4 & 0
\end{pmatrix}$$

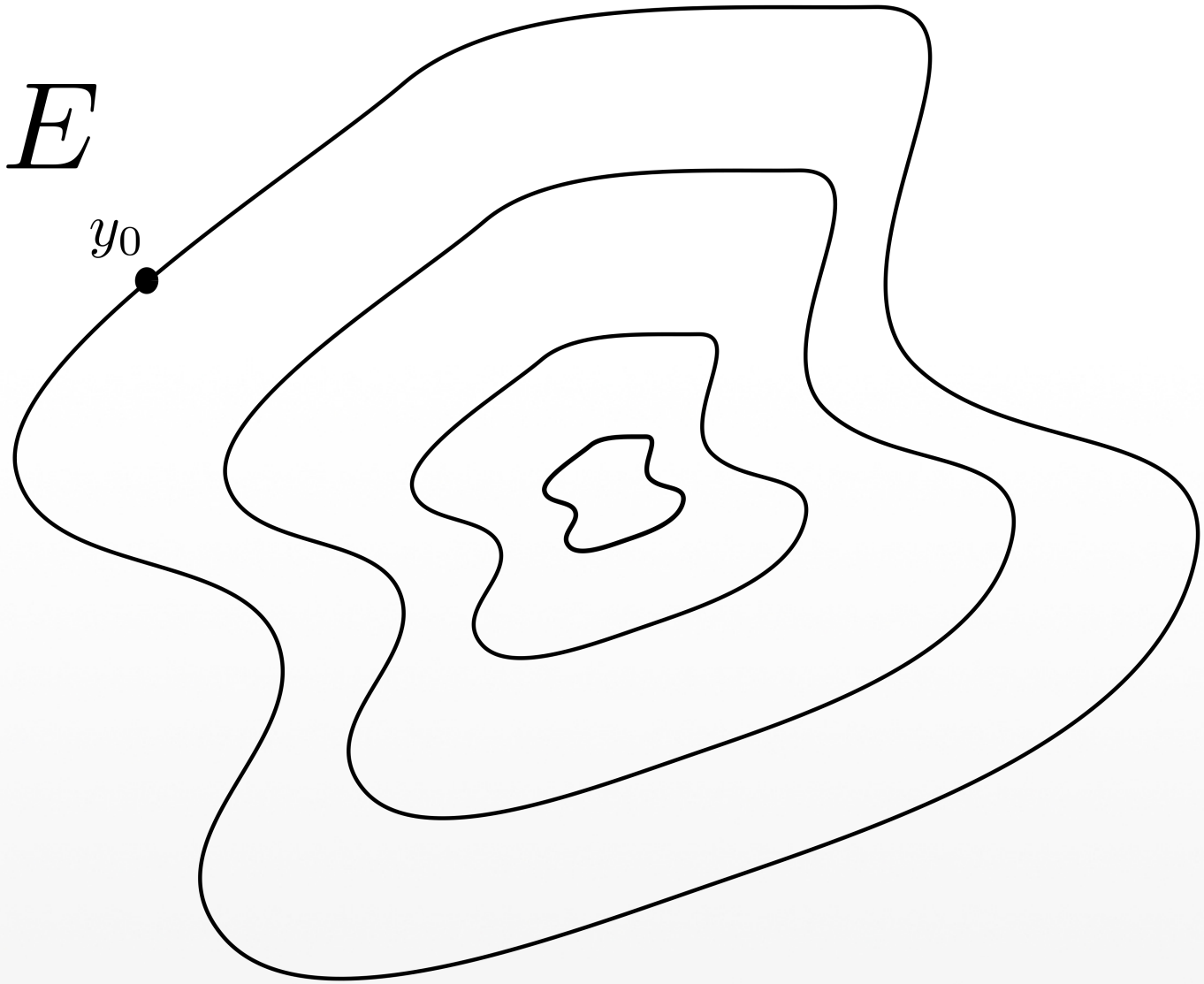
Parametric Gradient Flows



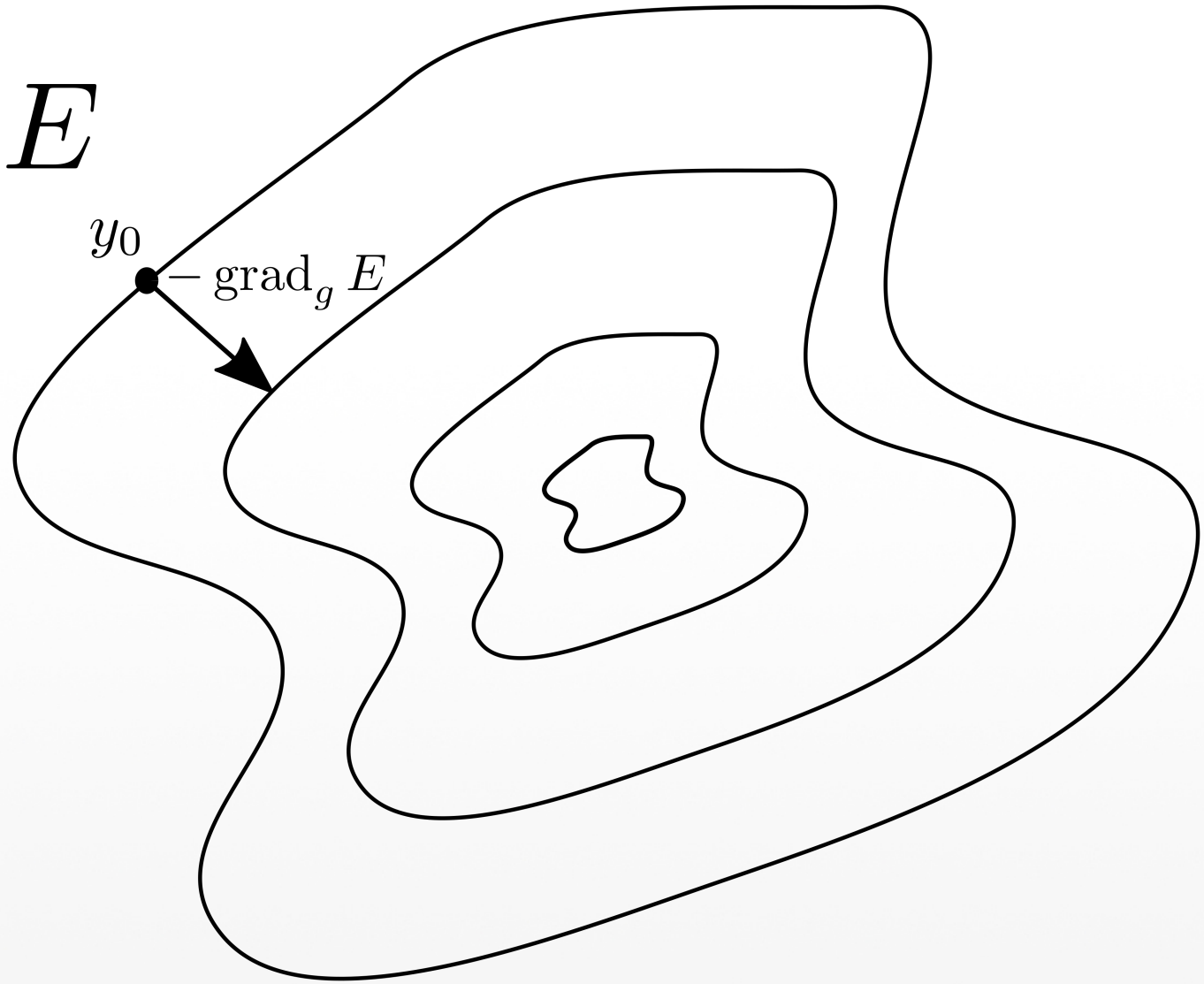
Gradient Flow



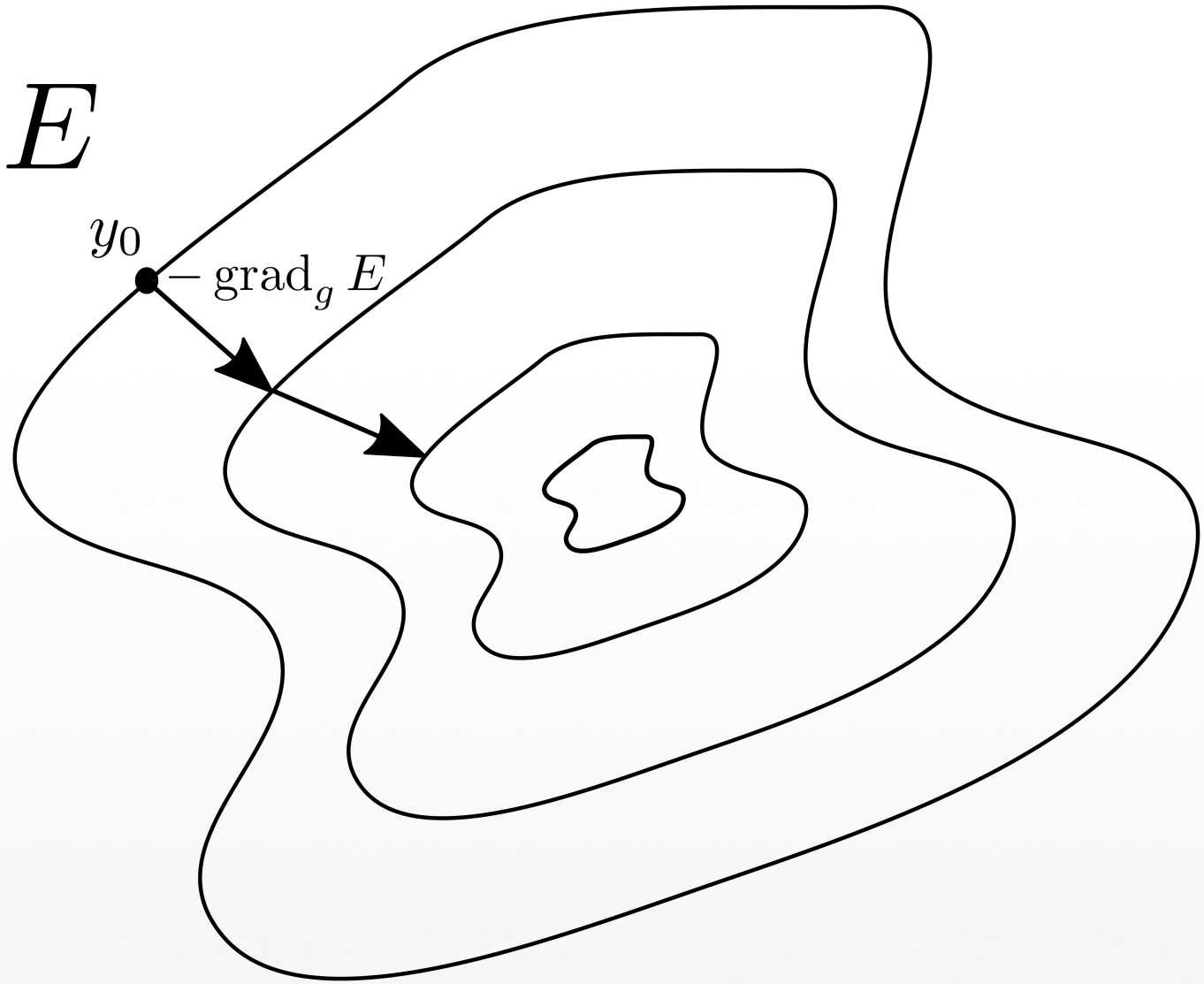
Gradient Flow



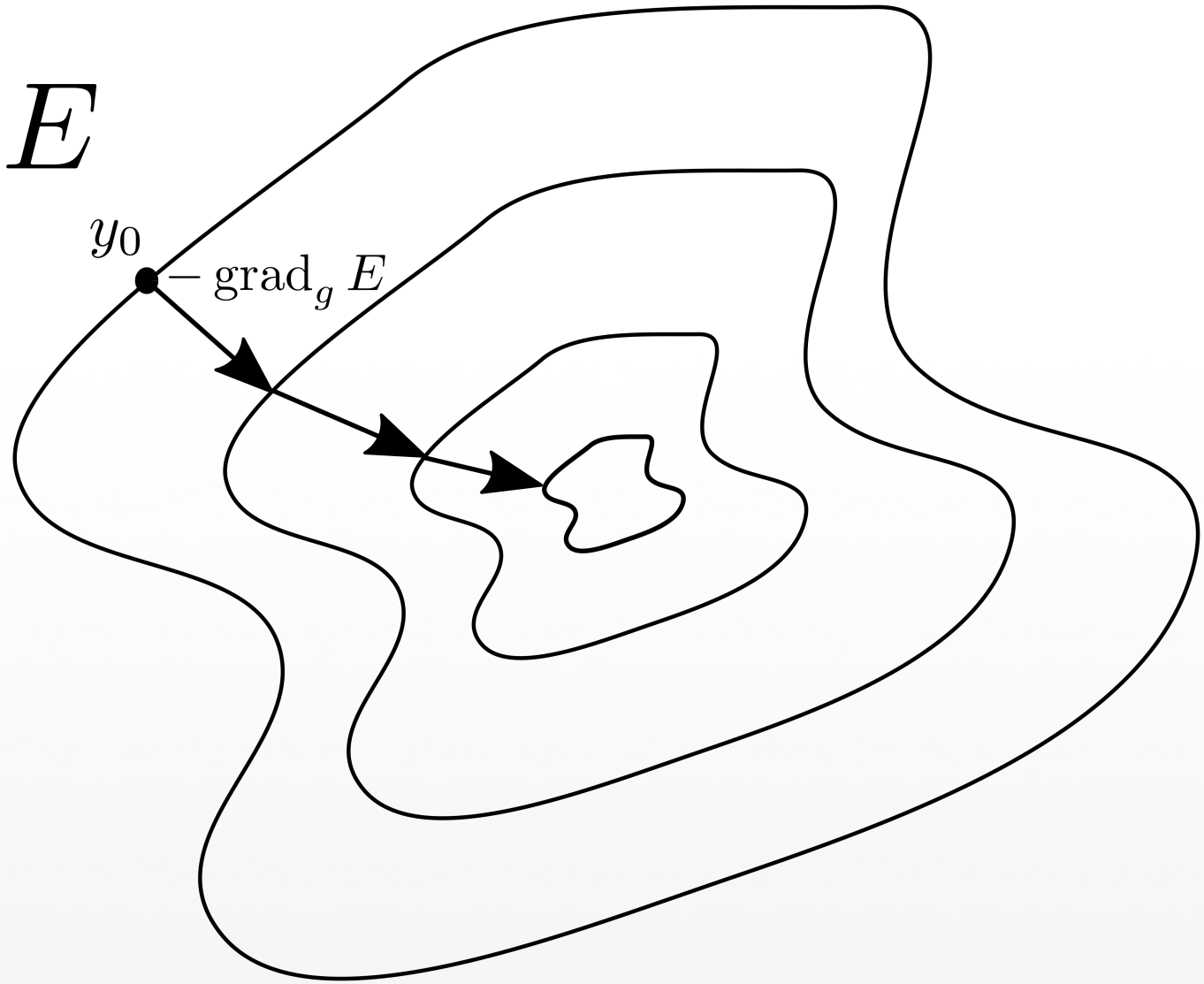
Gradient Flow



Gradient Flow

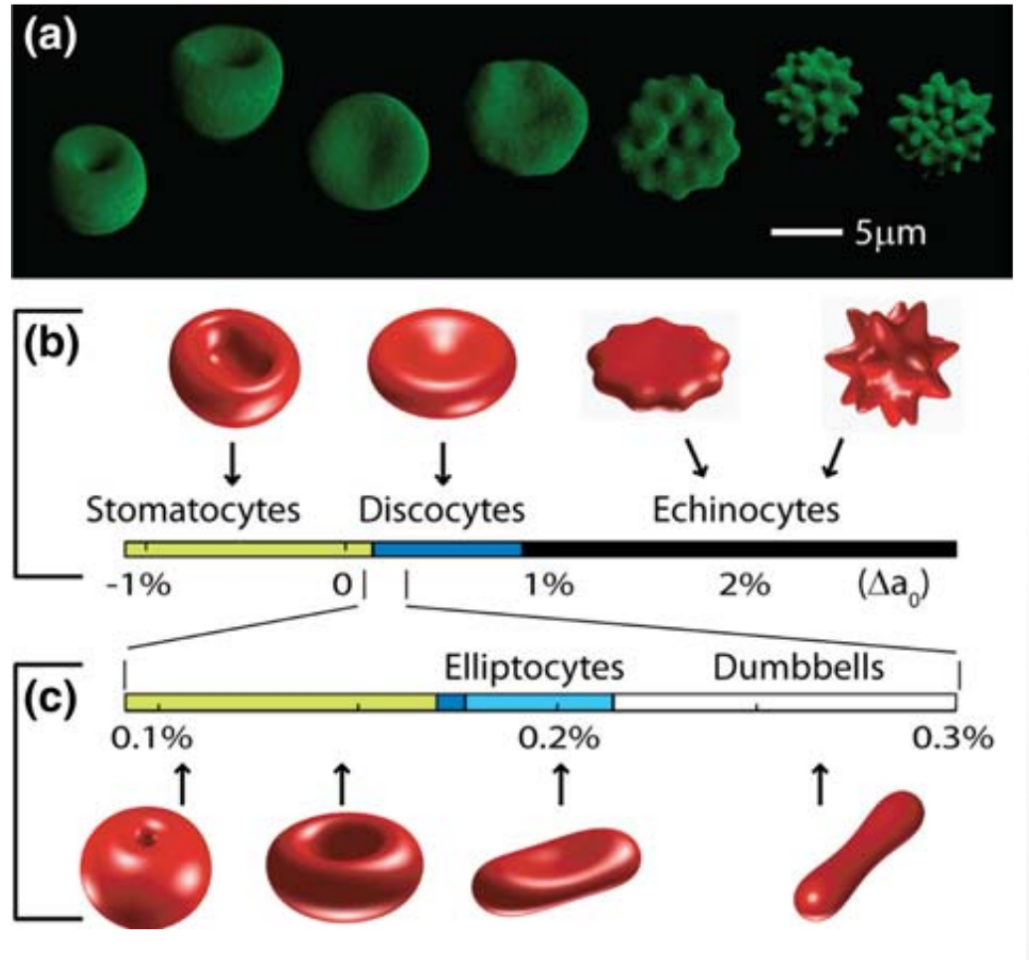


Gradient Flow



Biological Membranes

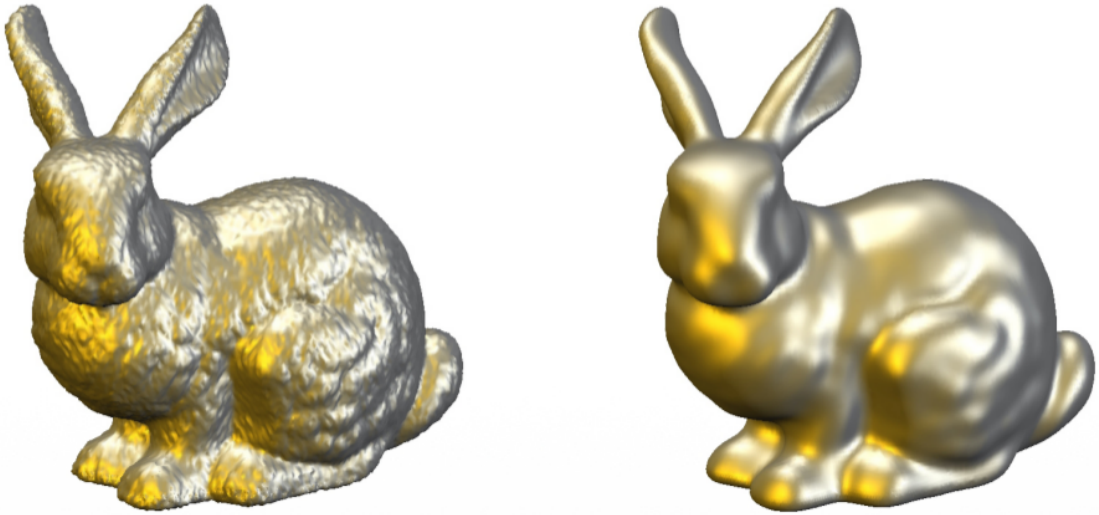
Computation of biomembranes



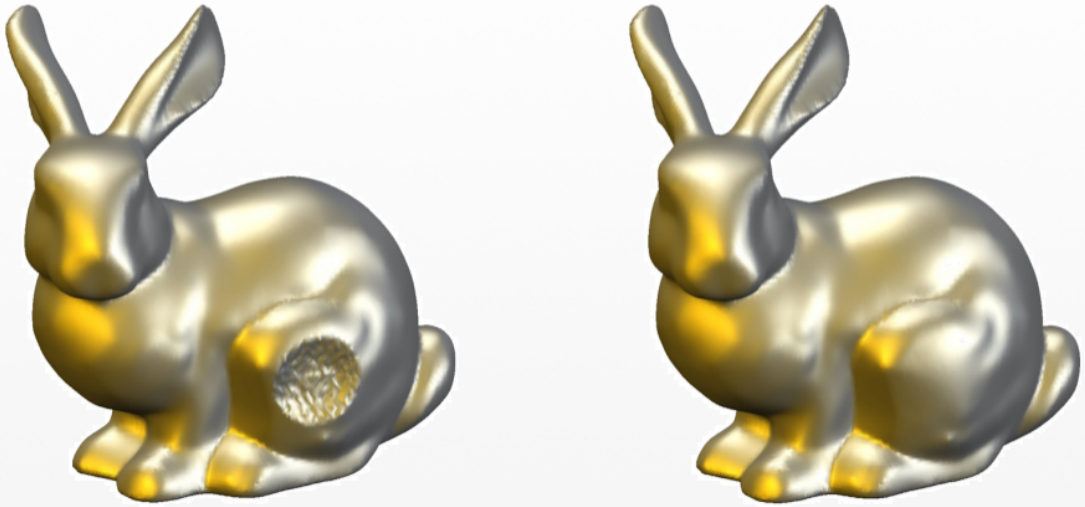
[Khairy et al 2008]

Surface Processing and Modeling

Surface denoising



Surface restoration



[Ohlischläger 2010]

Parametric Gradient Flow

Strong formulation

$$y_t = -\mathrm{grad}_g e[y]$$
$$0 = c[y]$$

Parametric Gradient Flow

Strong formulation

$$y_t = -\mathrm{grad}_g\, e[y]$$
$$0 = c[y]$$

Weak formulation

$$g(y_t, \varphi) = -e'[y](\varphi) \quad \forall \varphi$$
$$0 = c'[y](y_t)$$

Parametric Gradient Flow

Strong formulation

$$y_t = -\mathrm{grad}_g e[y]$$
$$0 = c[y]$$

Weak formulation

$$g(y_t, \varphi) = -e'[y](\varphi) \quad \forall \varphi$$
$$0 = c'[y](y_t)$$

Energy diminishing property

$$e[y(t)] \leq e[y(s)] \quad \forall t < s$$

Parametric Gradient Flow

Strong formulation

$$y_t = -\mathrm{grad}_g e[y]$$
$$0 = c[y]$$

Weak formulation

$$g(y_t, \varphi) = -e'[y](\varphi) \quad \forall \varphi$$
$$0 = c'[y](y_t)$$

Energy diminishing property

$$e[y(t)] \leq e[y(s)] \quad \forall t < s$$

Energies

Area

$$e[y] = \int_{\mathcal{M}} \mathrm{d}a$$

Parametric Gradient Flow

Strong formulation

$$y_t = -\mathrm{grad}_g\, e[y]$$
$$0 = c[y]$$

Weak formulation

$$g(y_t, \varphi) = -e'[y](\varphi) \quad \forall \varphi$$
$$0 = c'[y](y_t)$$

Energy diminishing property

$$e[y(t)] \leq e[y(s)] \quad \forall t < s$$

Energies

Area

$$e[y] = \int_{\mathcal{M}} \mathrm{d}a$$

Willmore

$$e[y] = \int_{\mathcal{M}} \mathbf{h}^2 \,\mathrm{d}a$$

Parametric Gradient Flow

Strong formulation

$$y_t = -\mathrm{grad}_g\, e[y]$$
$$0 = c[y]$$

Weak formulation

$$g(y_t, \varphi) = -e'[y](\varphi) \quad \forall \varphi$$
$$0 = c'[y](y_t)$$

Energy diminishing property

$$e[y(t)] \leq e[y(s)] \quad \forall t < s$$

Energies

Area

$$e[y] = \int_{\mathcal{M}} \mathrm{d}a$$

Willmore

$$e[y] = \int_{\mathcal{M}} \mathbf{h}^2 \,\mathrm{d}a$$

Metrics

L^2–metric: $\quad g(v, w) = \int_{\mathcal{M}} v \cdot w \,\mathrm{d}a$

Parametric Gradient Flow

Strong formulation

$$y_t = -\mathrm{grad}_g e[y]$$
$$0 = c[y]$$

Weak formulation

$$g(y_t, \varphi) = -e'[y](\varphi) \quad \forall \varphi$$
$$0 = c'[y](y_t)$$

Energy diminishing property

$$e[y(t)] \leq e[y(s)] \quad \forall t < s$$

Energies

Area

$$e[y] = \int_{\mathcal{M}} \mathrm{d}a$$

Willmore

$$e[y] = \int_{\mathcal{M}} \mathbf{h}^2 \, \mathrm{d}a$$

Metrics

L^2–metric: $g(v, w) = \int_{\mathcal{M}} v \cdot w \, \mathrm{d}a$

H^1–metric: $g(v, w) = \int_{\mathcal{M}} v \cdot w + \nabla_{\mathcal{M}} v \cdot \nabla_{\mathcal{M}} w \, \mathrm{d}a$

Parametric Gradient Flow

Strong formulation

$$y_t = -\mathrm{grad}_g e[y]$$
$$0 = c[y]$$

Weak formulation

$$g(y_t, \varphi) = -e'[y](\varphi) \quad \forall \varphi$$
$$0 = c'[y](y_t)$$

Energy diminishing property

$$e[y(t)] \leq e[y(s)] \quad \forall t < s$$

Energies

Area

$$e[y] = \int_{\mathcal{M}} \mathrm{d}a$$

Willmore

$$e[y] = \int_{\mathcal{M}} \mathbf{h}^2 \,\mathrm{d}a$$

Metrics

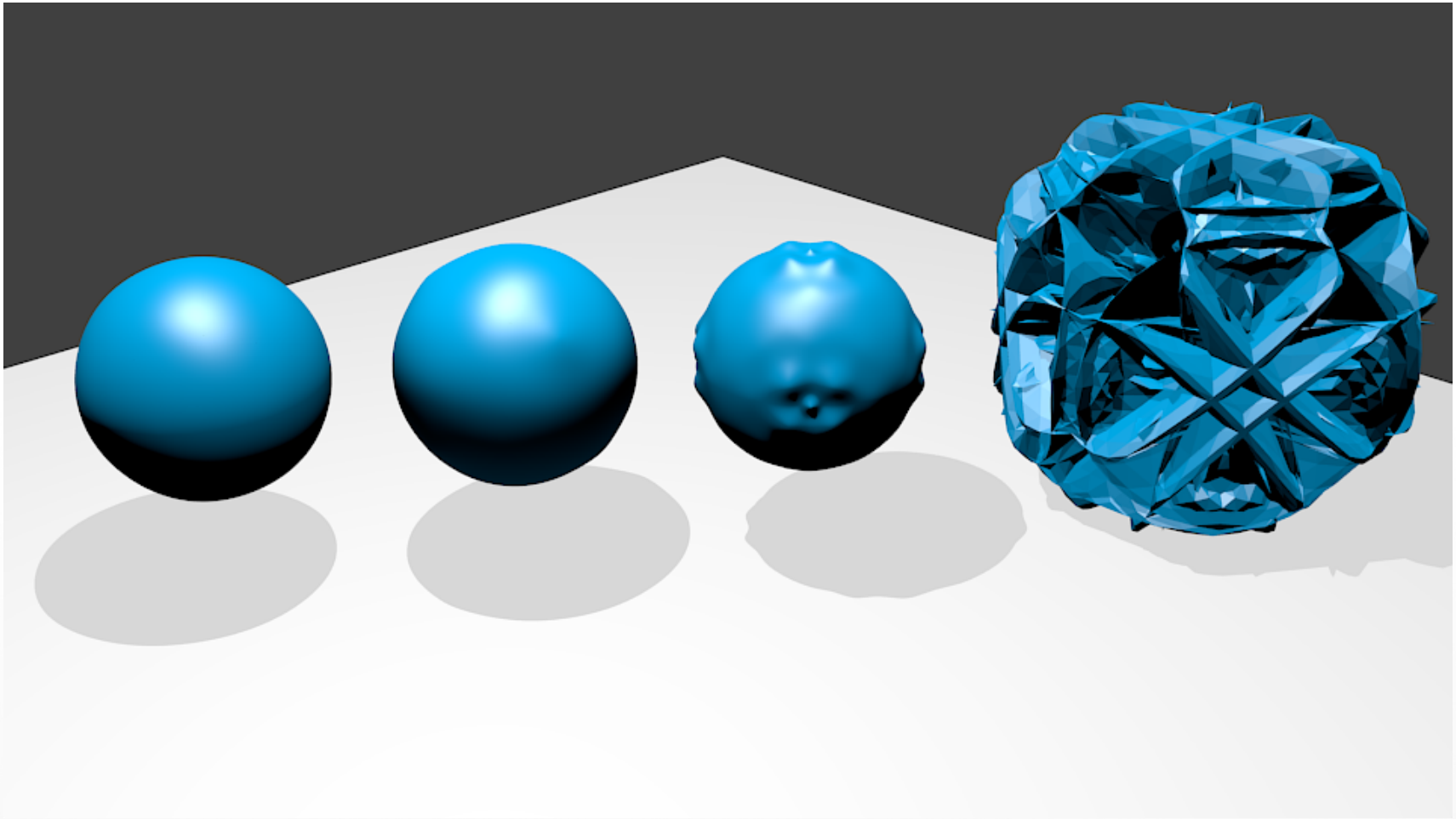
L^2**–metric:** $g(v, w) = \int_{\mathcal{M}} v \cdot w \,\mathrm{d}a$

H^1**–metric:** $g(v, w) = \int_{\mathcal{M}} v \cdot w + \nabla_{\mathcal{M}} v \cdot \nabla_{\mathcal{M}} w \,\mathrm{d}a$

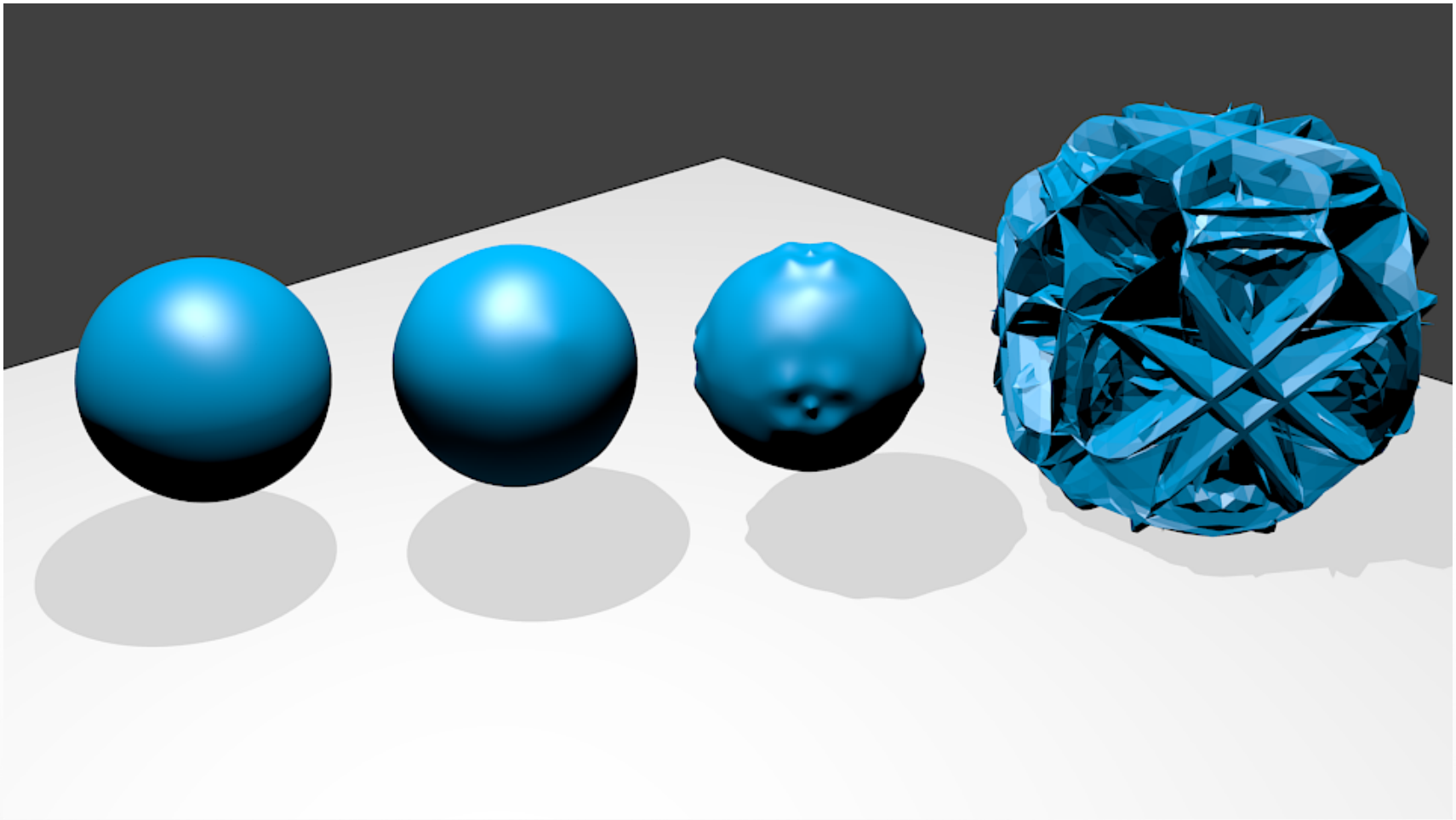
H^2_Δ**–metric:** $g(v, w) = \int_{\mathcal{M}} v \cdot w + \nabla_{\mathcal{M}} v \cdot \nabla_{\mathcal{M}} w + \Delta_{\mathcal{M}} v \cdot \Delta_{\mathcal{M}} w \,\mathrm{d}a$

Explicit Time Integrators

Explicit Time Integrators



Explicit Time Integrators



⇒Lack of stability

Implicit Time Integrators

Implicit Euler

$$\max_m \|y(t_m) - Y_m\| \leq C(\tau + h^p)$$

- Small time step $\tau = \mathcal{O}(h^p)$ or reduced convergence order

ODE: $\dot{y} = f(t, y),\ y(0) = y_0$

$$\Downarrow$$

$$y_{m+1} = y_m + \tau f(t_{m+1}, y_{m+1})\ m \geq 0,$$

where $t_{m+1} = t_m + \tau$

Implicit Time Integrators

Higher order implicit Runge-Kutta

$$\max_m \|y(t_m) - Y_m\| \leq C(\tau^p + h^p)$$

ODE: $\dot{y} = f(t, y),\ y(0) = y_0$

$$\Downarrow$$

$$y_{m+1} = y_m + \tau \sum_{i=1}^{s} b_i f(t_m + c_i\tau, y_{mi})\ m \geq 0$$

where $t_{m+1} = t_m + \tau$

and

$$y_{mi} = y_m + \tau \sum_{i=1}^{s} a_{ij} f(t_m + c_i\tau, y_{mj})\ i = 1, \ldots, s$$

Butcher tableau

c_1	a_{11}	$\ldots$	a_{1s}
$\vdots$	$\vdots$	$\ddots$	$\vdots$
c_s	a_{s1}	$\ldots$	a_{ss}
	b_1	$\ldots$	b_s

Implicit Time Integrators

Higher order implicit Runge-Kutta

$$\max_m \|y(t_m) - Y_m\| \leq C(\tau^p + h^p)$$

Gradient Flow: $\mathbf{G}_Y(\dot{Y}, \Psi) = -\mathbf{E}'[Y](\Psi) - \Lambda \mathbf{C}'[Y](\Psi), \quad \forall \Psi$

$$\mathbf{C}'[Y](\dot{Y}) = 0$$

$$\Downarrow$$

$$Y_{m+1} = Y_m + \tau \sum_{i=1}^{s} b_i \dot{Y}_{mi}$$

$$Y_{mi} = Y_m + \tau \sum_{j=1}^{s} a_{ij} \dot{Y}_{mj}, \qquad i = 1, \ldots, s,$$

where the internal stages satisfy

$$\mathbf{G}_{Y_{mi}}(\dot{Y}_{mi}, \Psi) = -\mathbf{E}'[Y_{mi}](\Psi) - \Lambda_i \mathbf{C}'[Y_{mi}](\Psi) \quad \forall\ \Psi, \quad i = 1, \ldots, s,$$

$$\mathbf{C}'[Y_{mi}](\dot{Y}_{mi}) = 0$$

Butcher tableau

c_1	a_{11}	$\ldots$	a_{1s}
$\vdots$	$\vdots$	$\ddots$	$\vdots$
c_s	a_{s1}	$\ldots$	a_{ss}
	b_1	$\ldots$	b_s

Implicit Time Integrators

Higher order implicit Runge-Kutta

$$\max_m \|y(t_m) - Y_m\| \leq C(\tau^p + h^p)$$

Theorem

$$\mathbf{E}[Y_{m+1}] \leq \mathbf{E}[Y_m] \quad m \geq 0$$

Gradient Flow: $\mathbf{G}_Y(\dot{Y}, \Psi) = -\mathbf{E}'[Y](\Psi) - \Lambda \mathbf{C}'[Y](\Psi), \quad \forall \Psi$

$$\mathbf{C}'[Y](\dot{Y}) = 0$$

$$\Downarrow$$

$$Y_{m+1} = Y_m + \tau \sum_{i=1}^{s} b_i \dot{Y}_{mi}$$

$$Y_{mi} = Y_m + \tau \sum_{j=1}^{s} a_{ij} \dot{Y}_{mj}, \qquad i = 1, \ldots, s,$$

where the internal stages satisfy

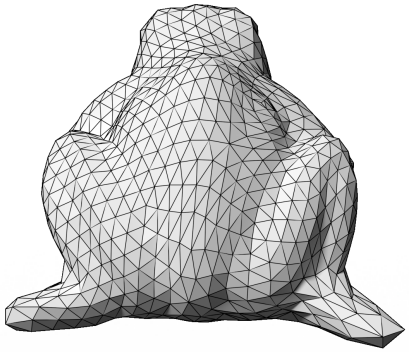
$$\mathbf{G}_{Y_{mi}}(\dot{Y}_{mi}, \Psi) = -\mathbf{E}'[Y_{mi}](\Psi) - \Lambda_i \mathbf{C}'[Y_{mi}](\Psi) \quad \forall\ \Psi, \quad i = 1, \ldots, s,$$

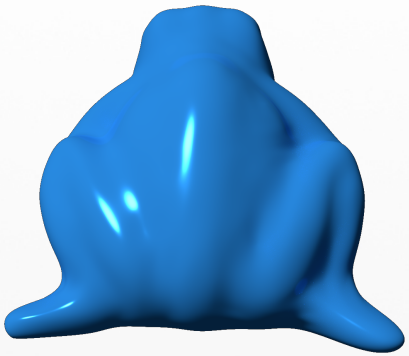
$$\mathbf{C}'[Y_{mi}](\dot{Y}_{mi}) = 0$$

Butcher tableau

c_1	a_{11}	$\ldots$	a_{1s}
$\vdots$	$\vdots$	$\ddots$	$\vdots$
c_s	a_{s1}	$\ldots$	a_{ss}
	b_1	$\ldots$	b_s

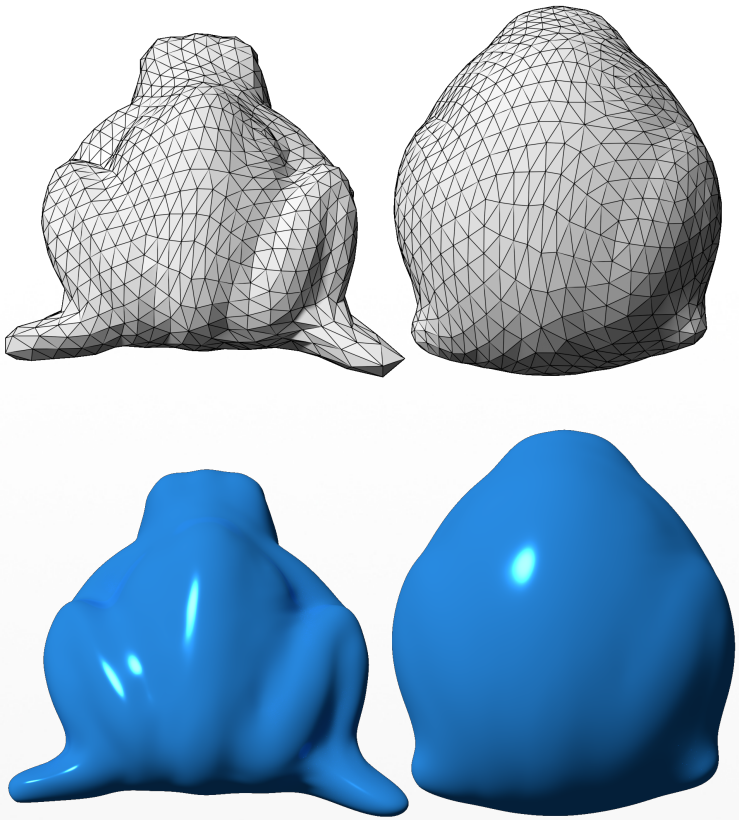
H^2_Δ–Willmore flow





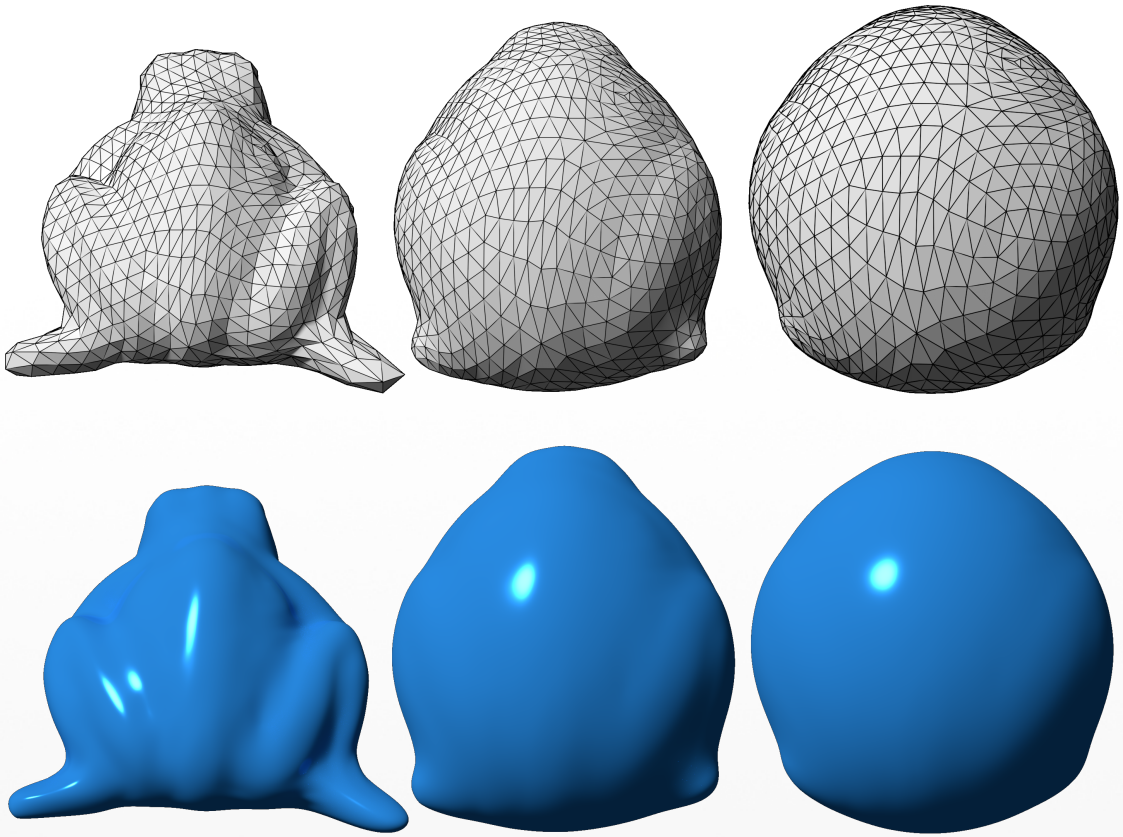
Initial

H^2_Δ–Willmore flow



Initial Step 1

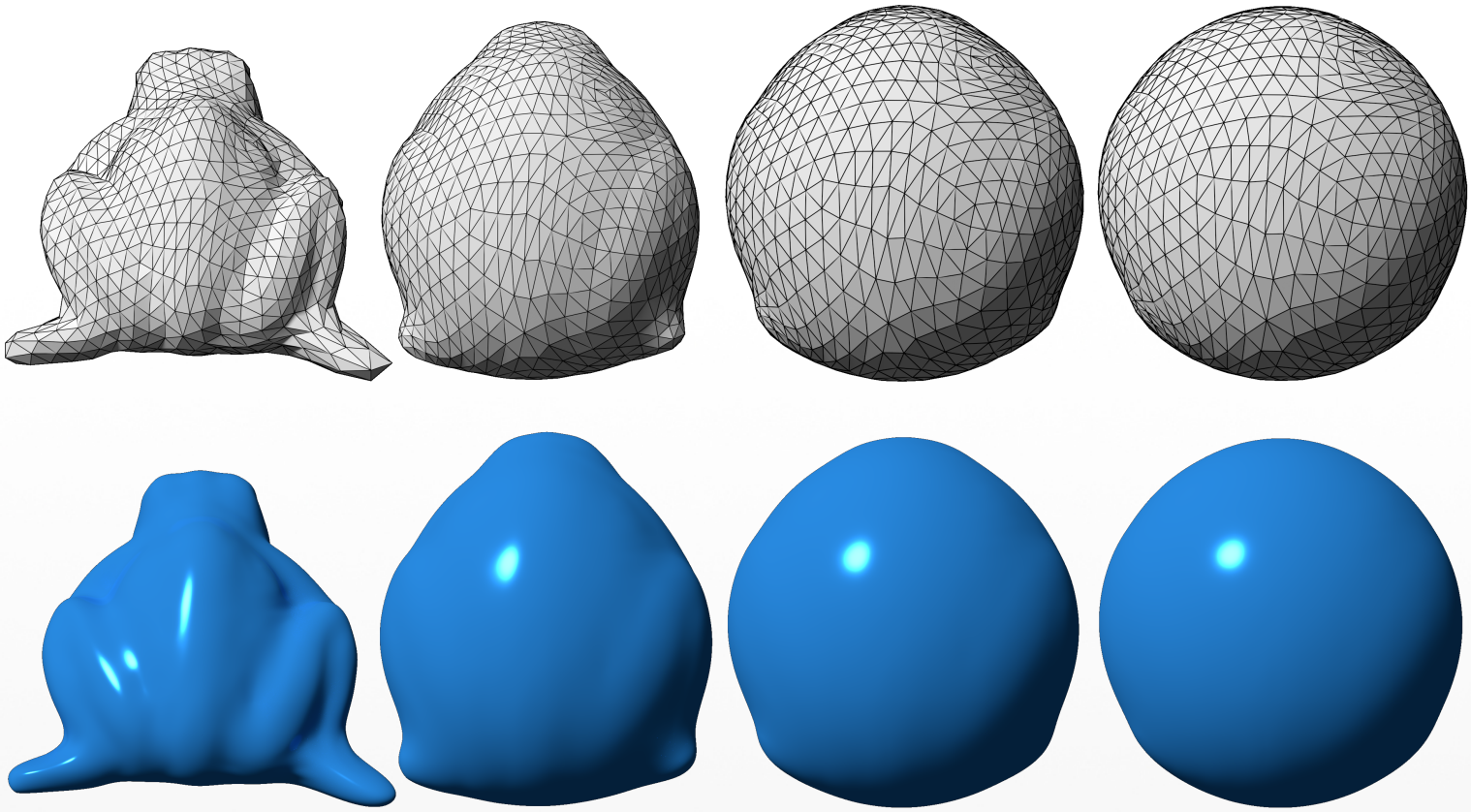
H^2_Δ–Willmore flow



Initial Step 1 Step 2

Results

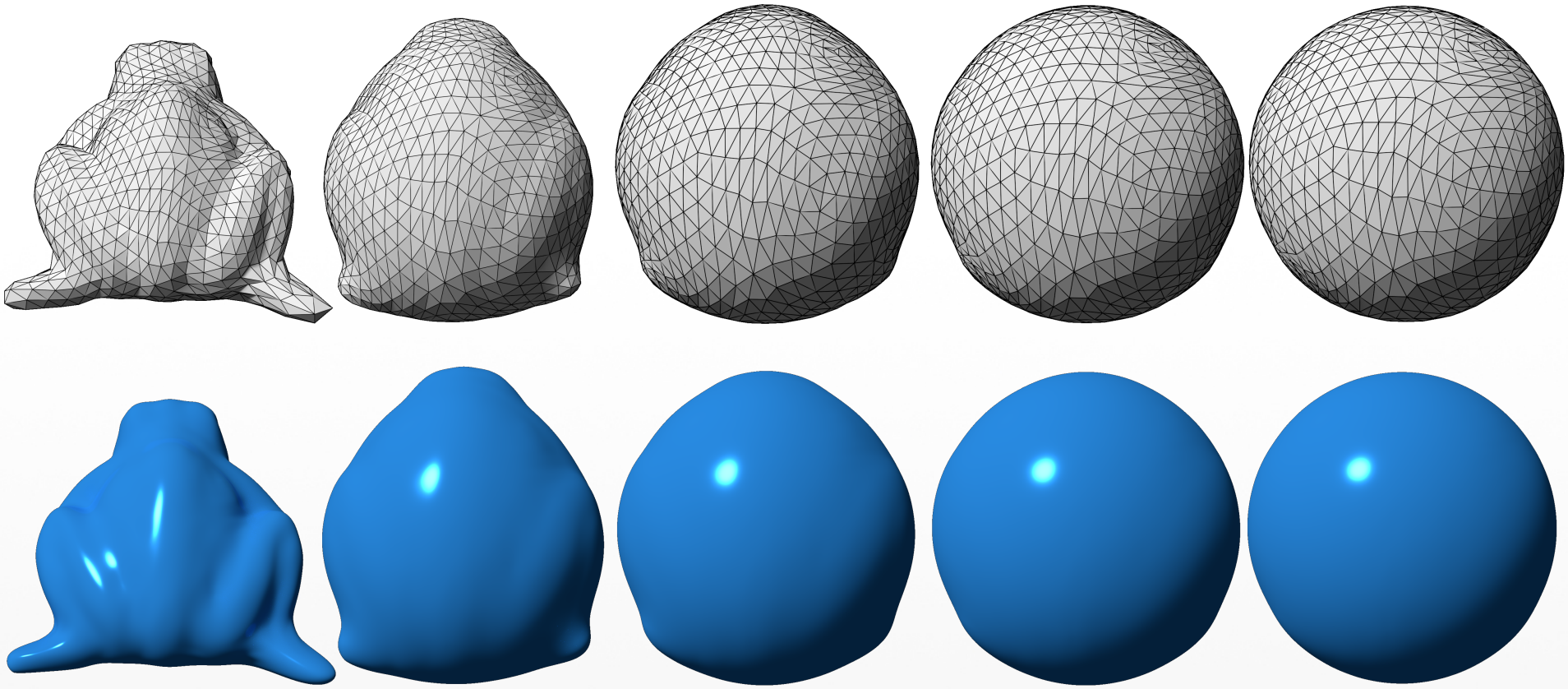
H^2_Δ–Willmore flow



Initial Step 1 Step 2 Step 3

Results

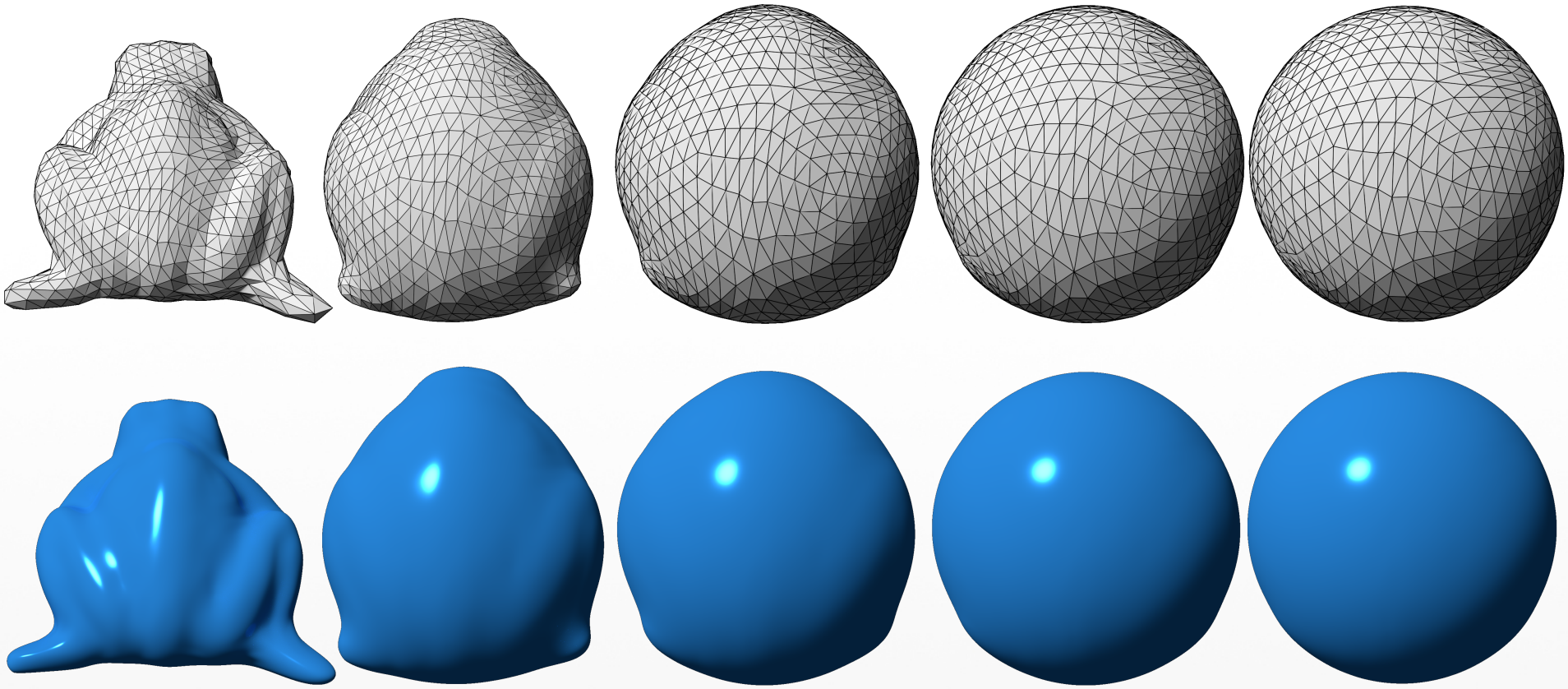
H^2_Δ–Willmore flow



Initial Step 1 Step 2 Step 3 Step 4

Results

H^2_Δ–Willmore flow

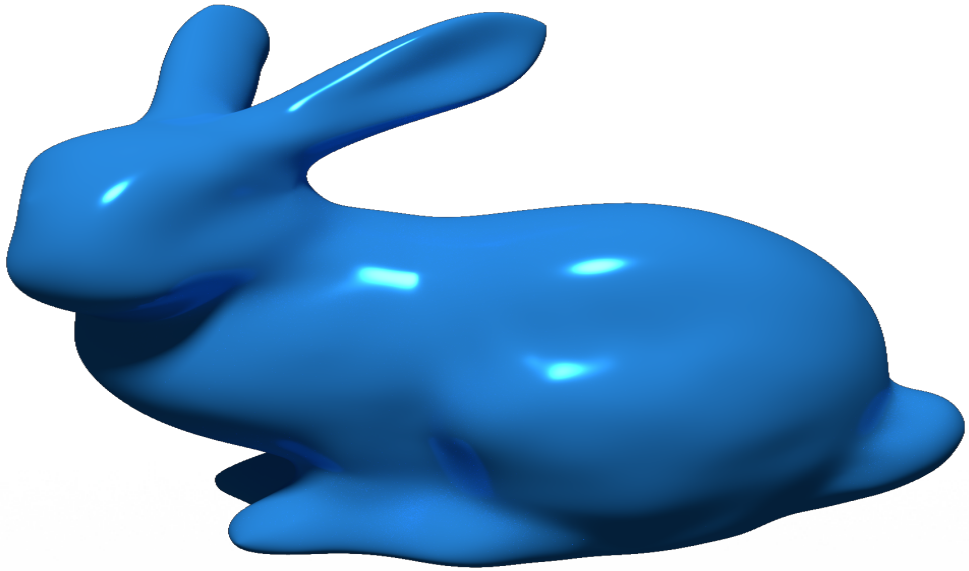


Initial | Step 1 | Step 2 | Step 3 | Step 4

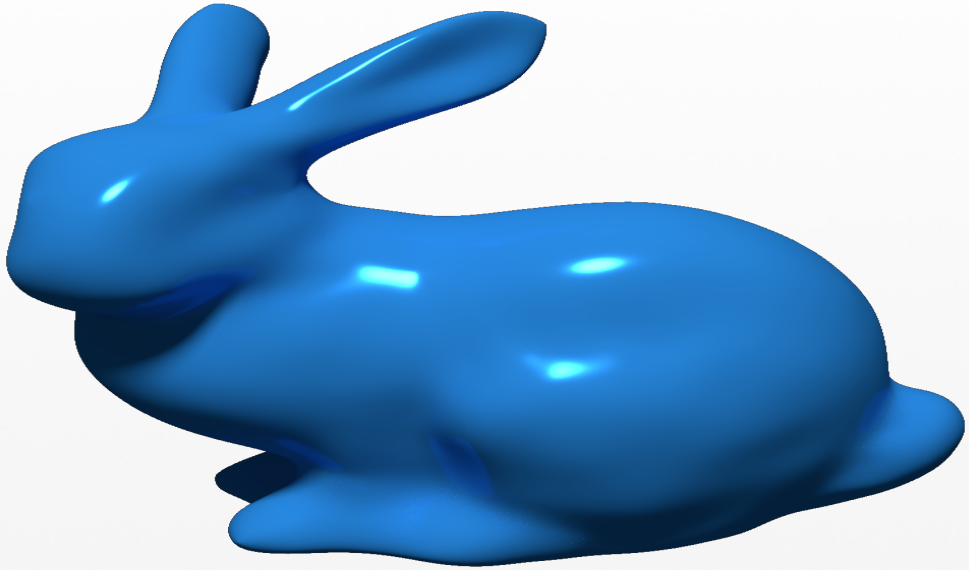
Large Time Steps: $\tau = 0.75,\ h = 0.11$

Results

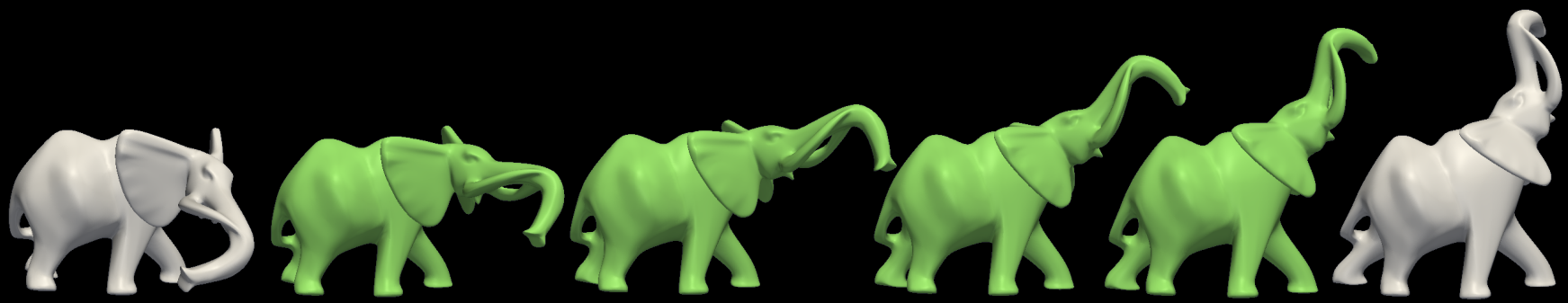
H^1–Willmore flow



H^2_Δ–Willmore flow



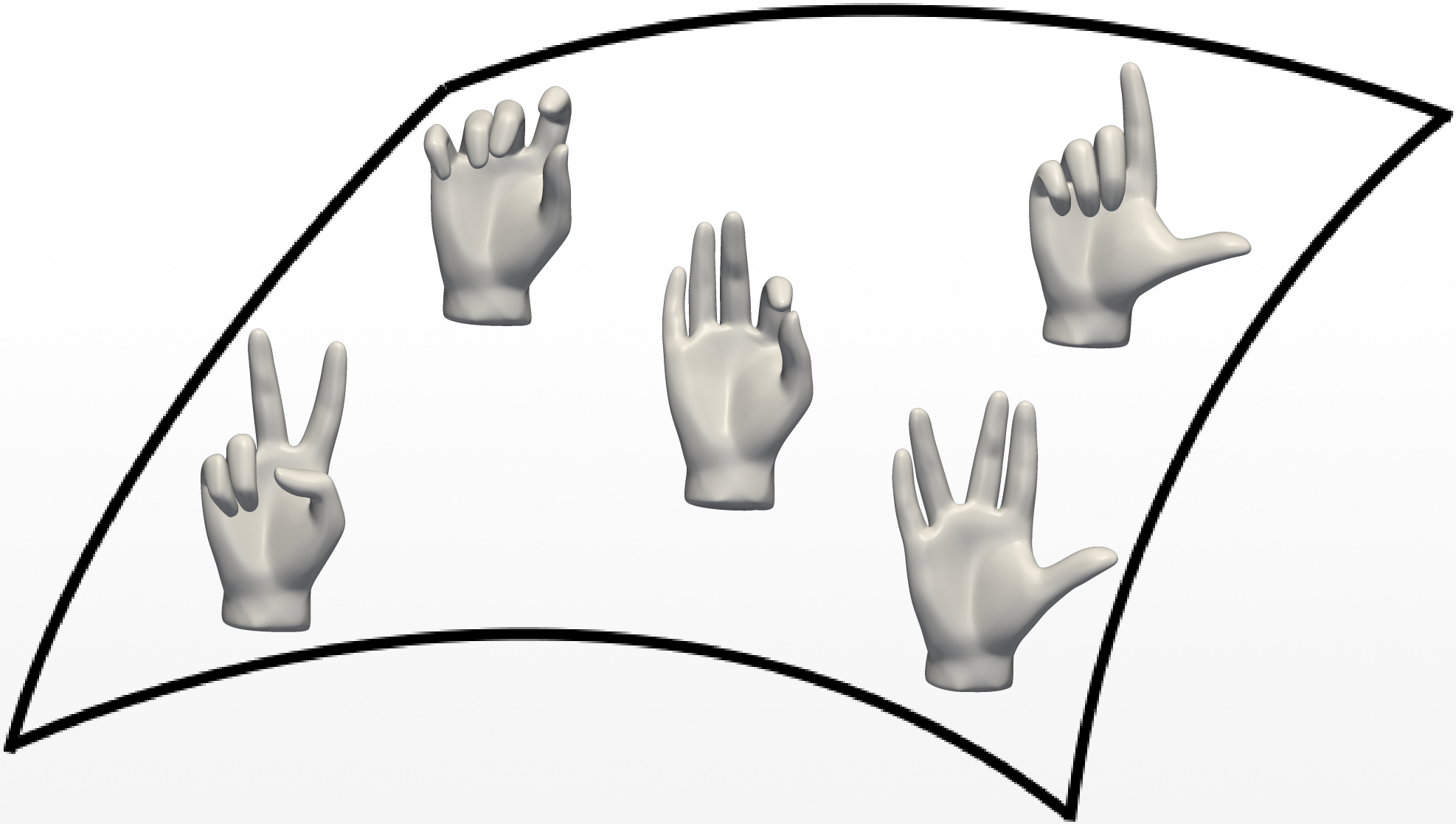
Bézier curves and Splines in Shape Space



Shape Space

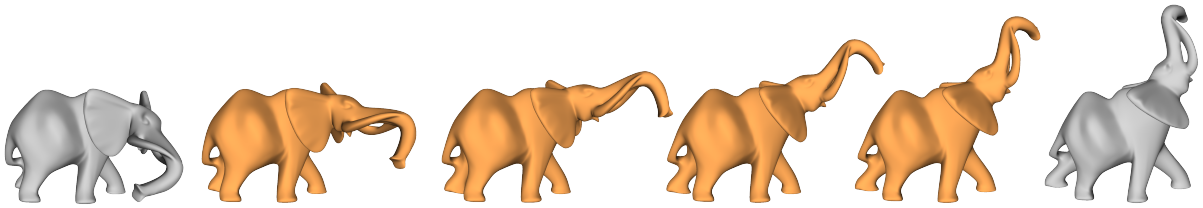
Shape Space

Shell space as Riemannian manifold

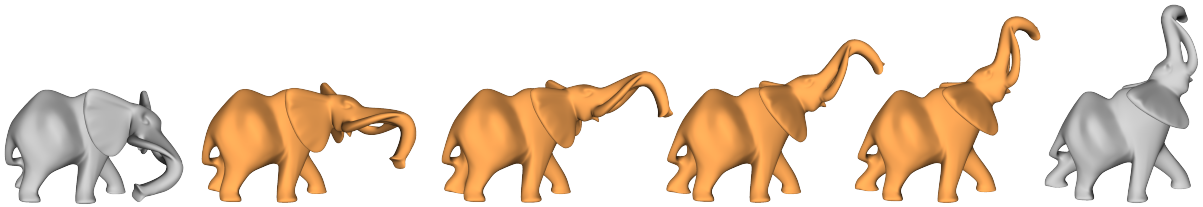


Geodesic Calculus

Geodesic Calculus

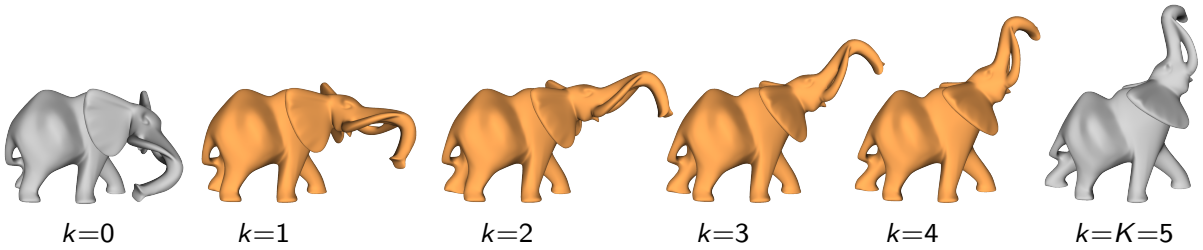


Geodesic Calculus



[Heeren et al 2012, 2014] developed a **time discrete geodesic calculus in shell space**

Geodesic Calculus

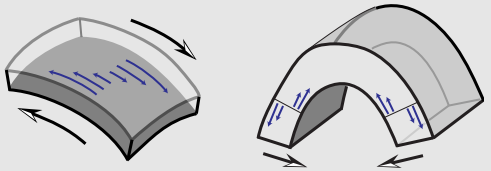


[Heeren et al 2012, 2014] developed a **time discrete geodesic calculus in shell space**

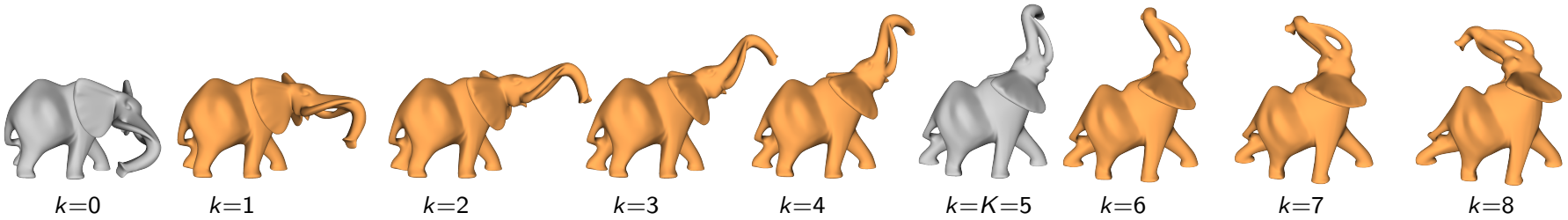
Discrete geodesics are (local) minimizers of the discrete path energy

$$\mathcal{E}[y_0, \ldots, y_K] = K \sum_{k=1}^{K} \mathcal{W}[y_{k-1}, y_k]$$

$$\mathrm{dist}^2(y_{k-1}, y_k) \approx \mathcal{W}[y_{k-1}, y_k] = \delta\mathcal{W}_{\mathrm{tang}}[y_{k-1}, y_k] + \delta^3\mathcal{W}_{\mathrm{bend}}[y_{k-1}, y_k]$$

tangential distortion  **bending**

Geodesic Calculus

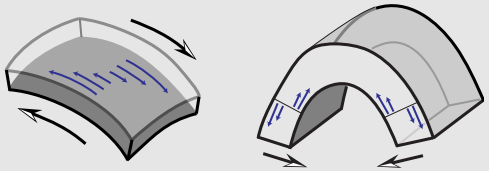


[Heeren et al 2012, 2014] developed a **time discrete geodesic calculus in shell space**

Discrete geodesics are (local) minimizers of the discrete path energy

$$\mathcal{E}[y_0, \ldots, y_K] = K \sum_{k=1}^{K} \mathcal{W}[y_{k-1}, y_k]$$

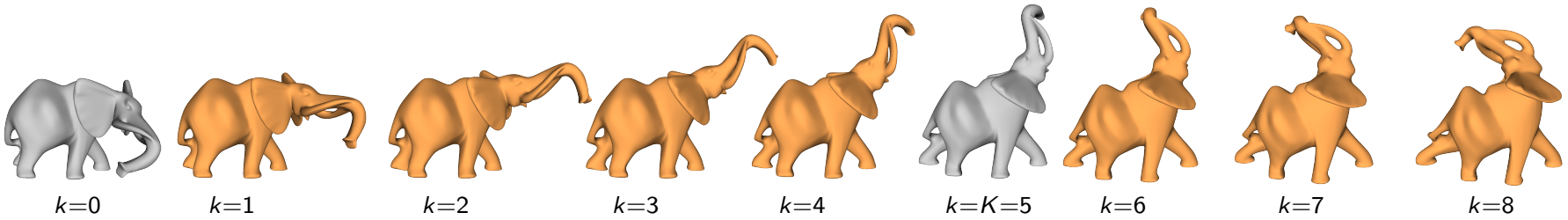
$$\mathrm{dist}^2(y_{k-1}, y_k) \approx \mathcal{W}[y_{k-1}, y_k] = \delta\mathcal{W}_{\mathrm{tang}}[y_{k-1}, y_k] + \delta^3\mathcal{W}_{\mathrm{bend}}[y_{k-1}, y_k]$$

tangential distortion  **bending**

Discrete exponential map defined variationally based on short discrete geodesics

$$y_k = \arg\min_{y} 2(\mathcal{W}[y_{k-2}, y_{k-1}] + \mathcal{W}[y_{k-1}, y])$$

Geodesic Calculus

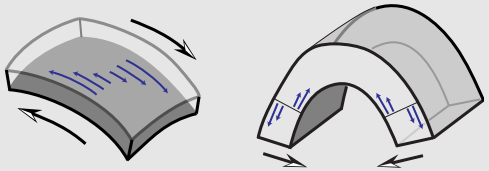


[Heeren et al 2012, 2014] developed a **time discrete geodesic calculus in shell space**

Discrete geodesics are (local) minimizers of the discrete path energy

$$\mathcal{E}[y_0, \ldots, y_K] = K \sum_{k=1}^{K} \mathcal{W}[y_{k-1}, y_k]$$

$$\mathrm{dist}^2(y_{k-1}, y_k) \approx \mathcal{W}[y_{k-1}, y_k] = \delta\mathcal{W}_{\mathrm{tang}}[y_{k-1}, y_k] + \delta^3\mathcal{W}_{\mathrm{bend}}[y_{k-1}, y_k]$$

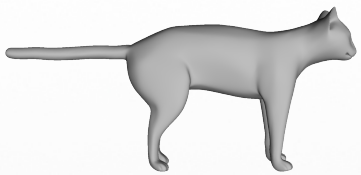
tangential distortion  **bending**

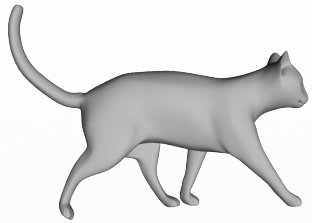
Discrete exponential map defined variationally based on short discrete geodesics

$$y_k = \arg\min_{y} 2(\mathcal{W}[y_{k-2}, y_{k-1}] + \mathcal{W}[y_{k-1}, y])$$

Discrete logarithm defined as initial displacements of shortest discrete geodesics

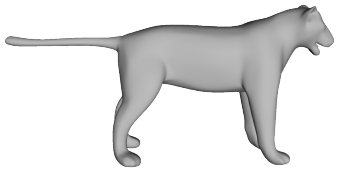
Geodesic Calculus

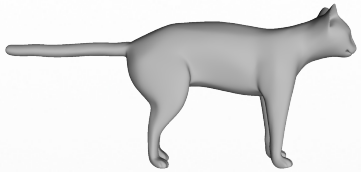


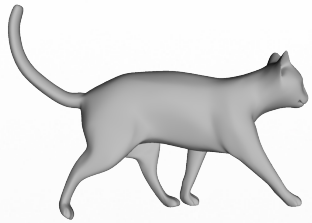


Discrete parallel transport defined via the construction of discrete Riemannian parallelograms

Geodesic Calculus

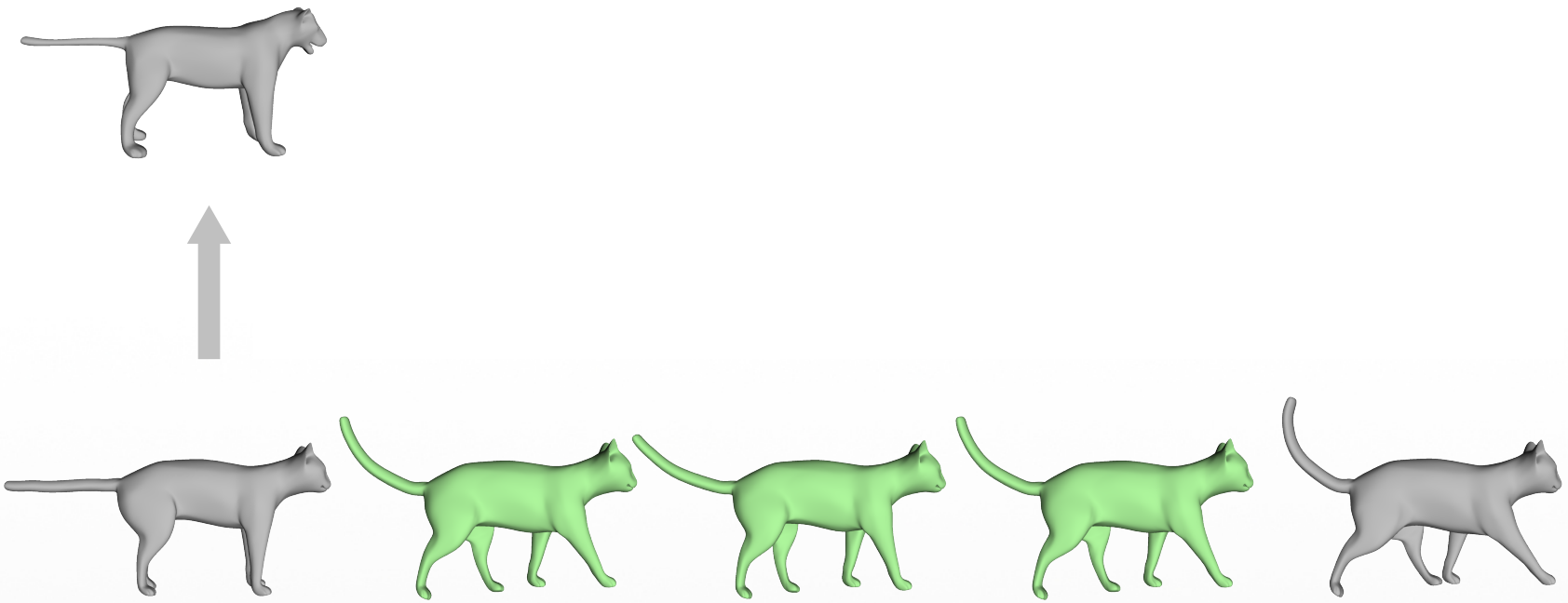






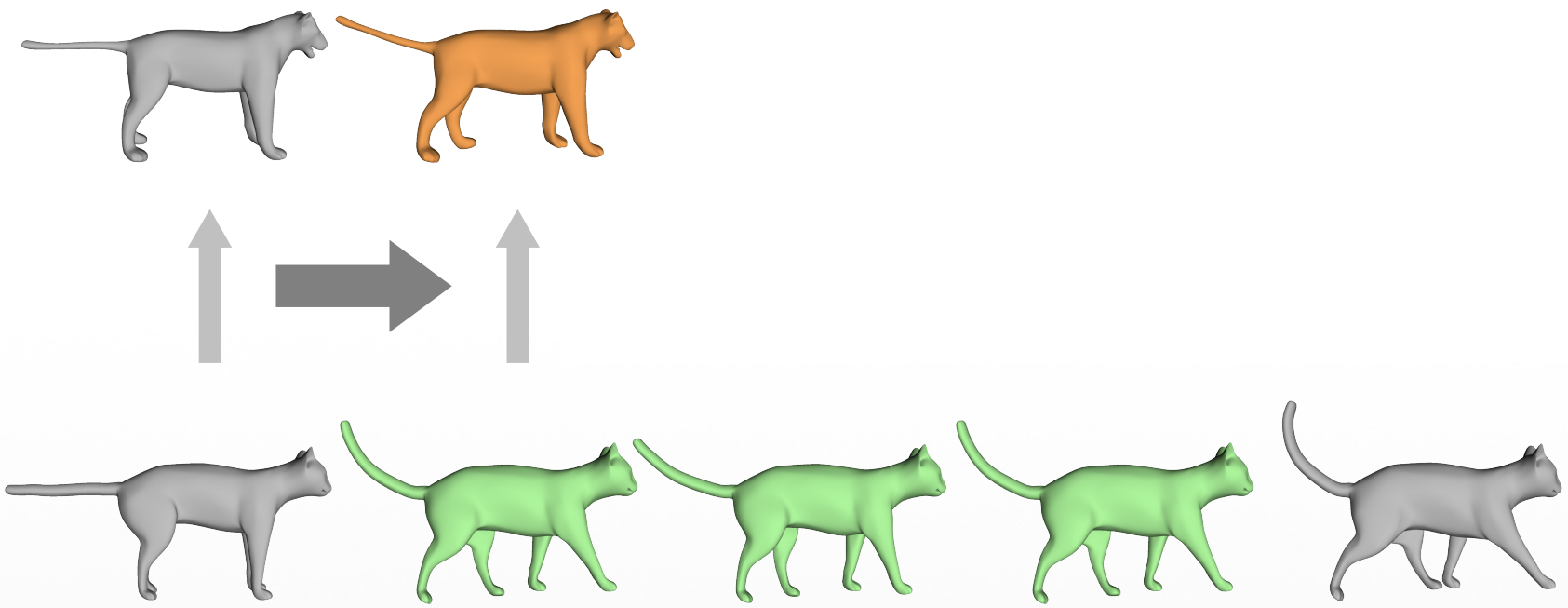
Discrete parallel transport defined via the construction of discrete Riemannian parallelograms

Geodesic Calculus



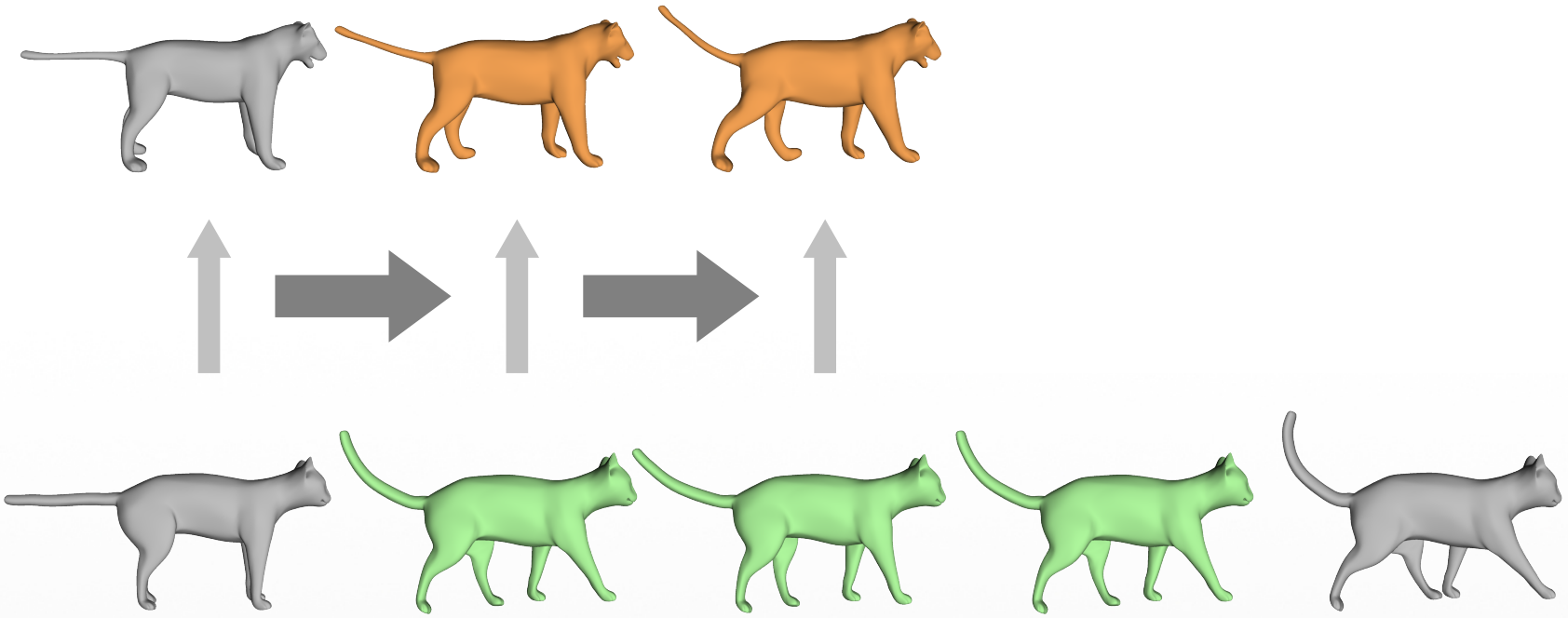
Discrete parallel transport defined via the construction of discrete Riemannian parallelograms

Geodesic Calculus



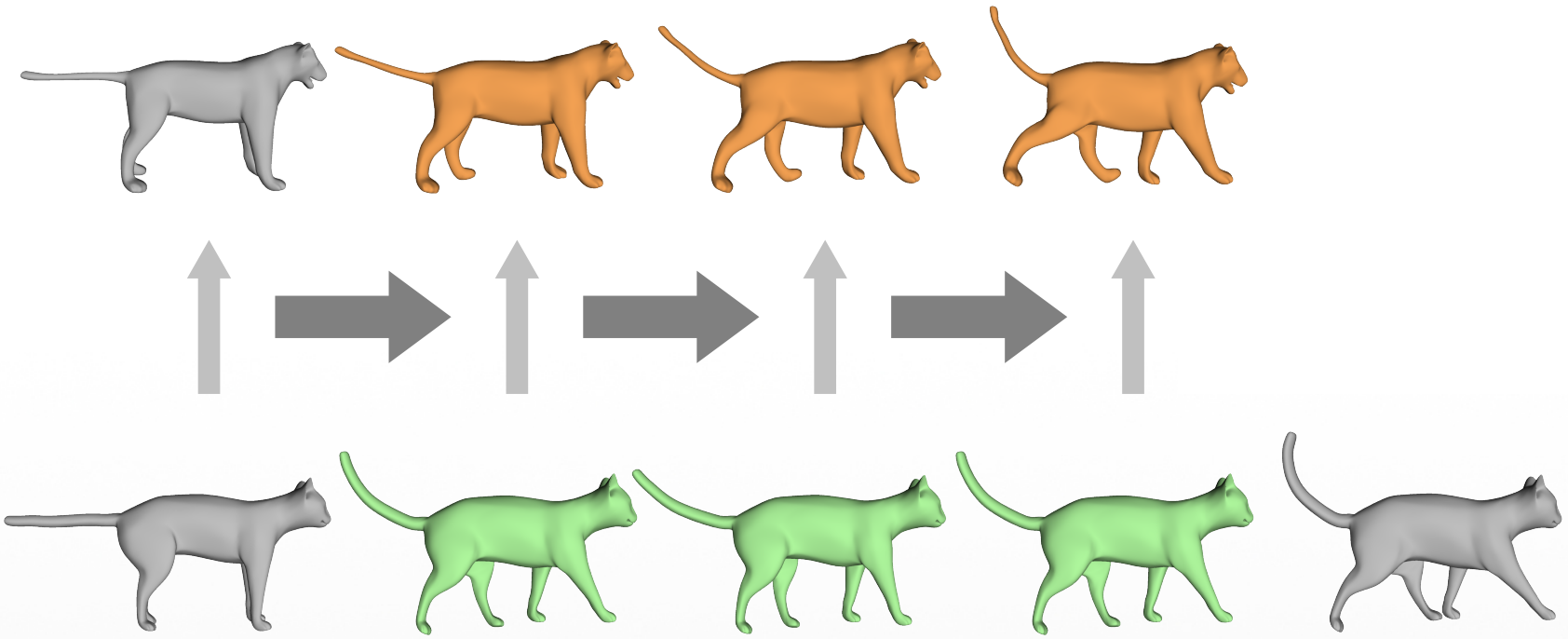
Discrete parallel transport defined via the construction of discrete Riemannian parallelograms

Geodesic Calculus



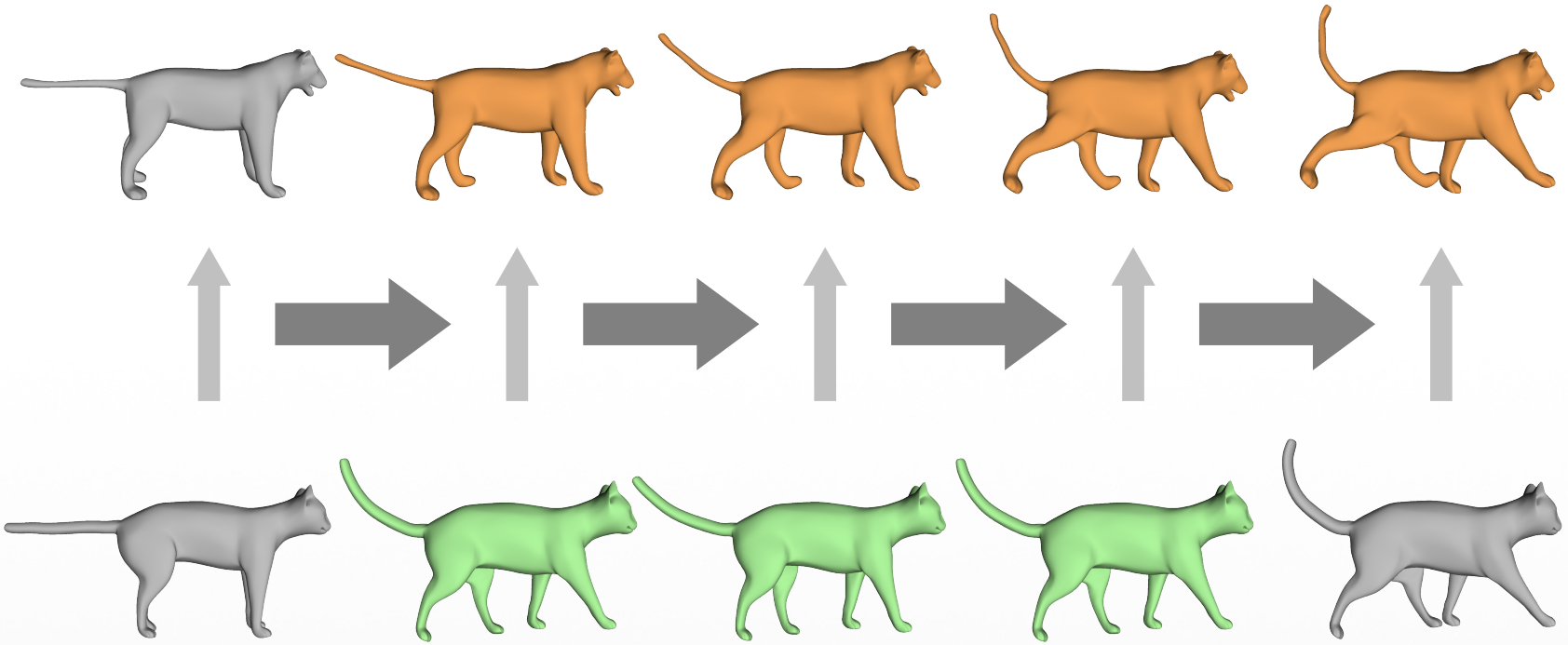
Discrete parallel transport defined via the construction of discrete Riemannian parallelograms

Geodesic Calculus



Discrete parallel transport defined via the construction of discrete Riemannian parallelograms

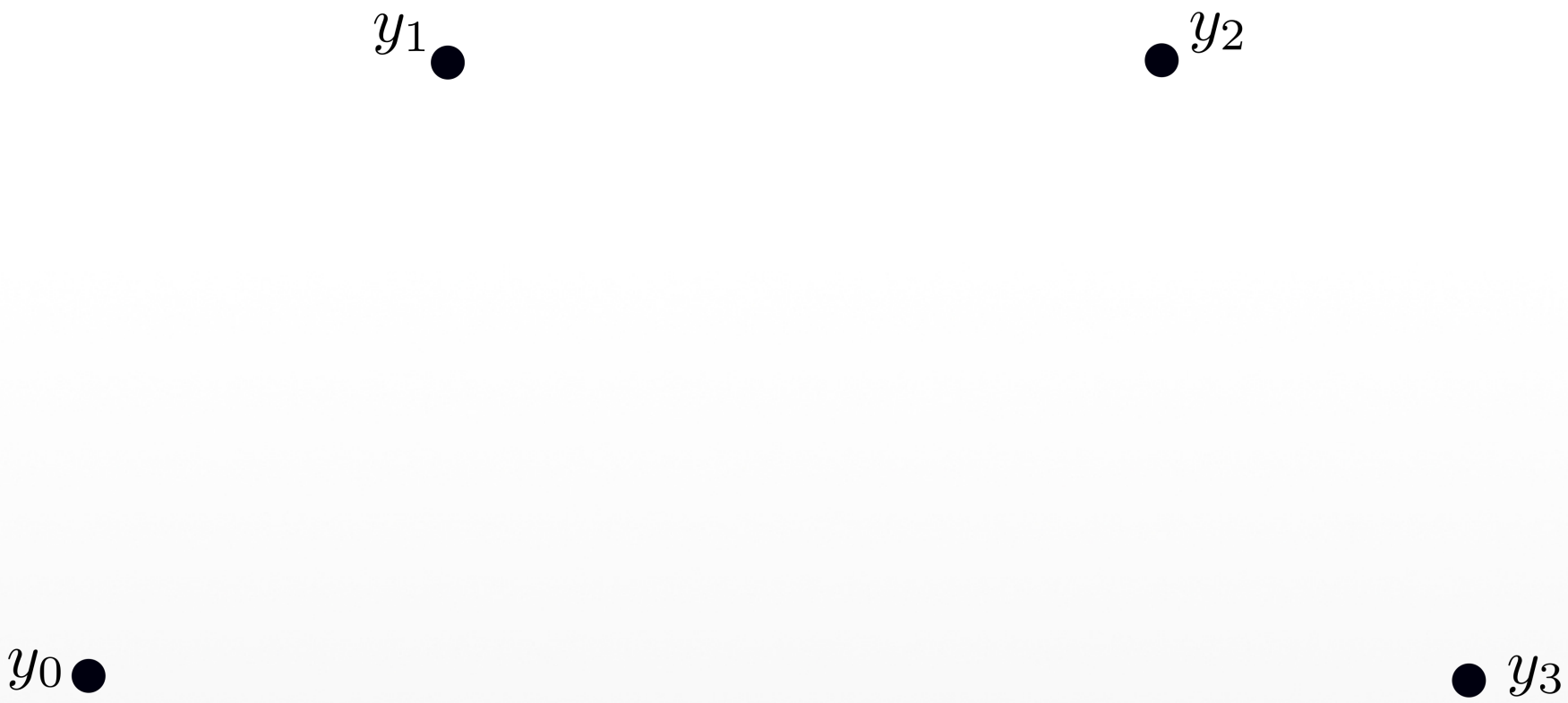
Geodesic Calculus



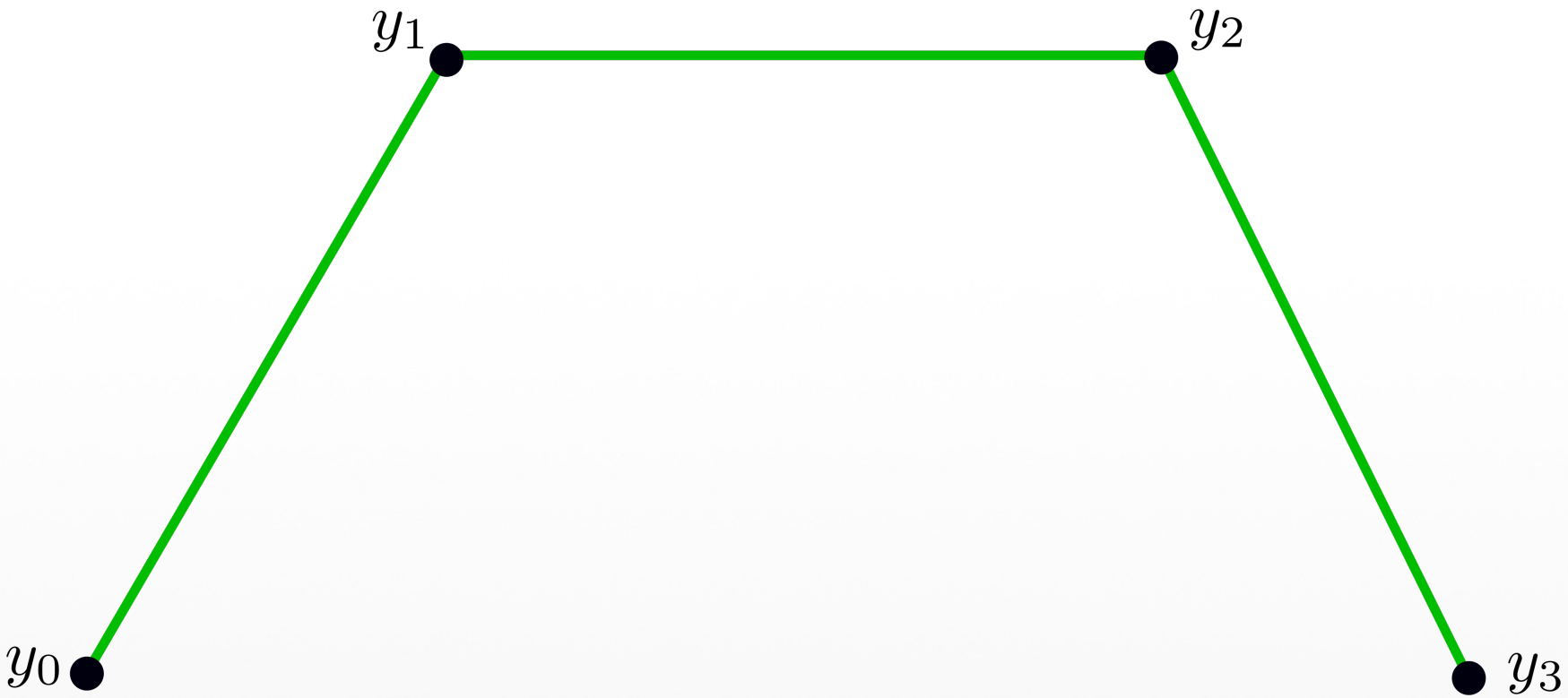
Discrete parallel transport defined via the construction of discrete Riemannian parallelograms

Bézier curves/De Casteljau Algorithm in $\mathbb{R}^n$

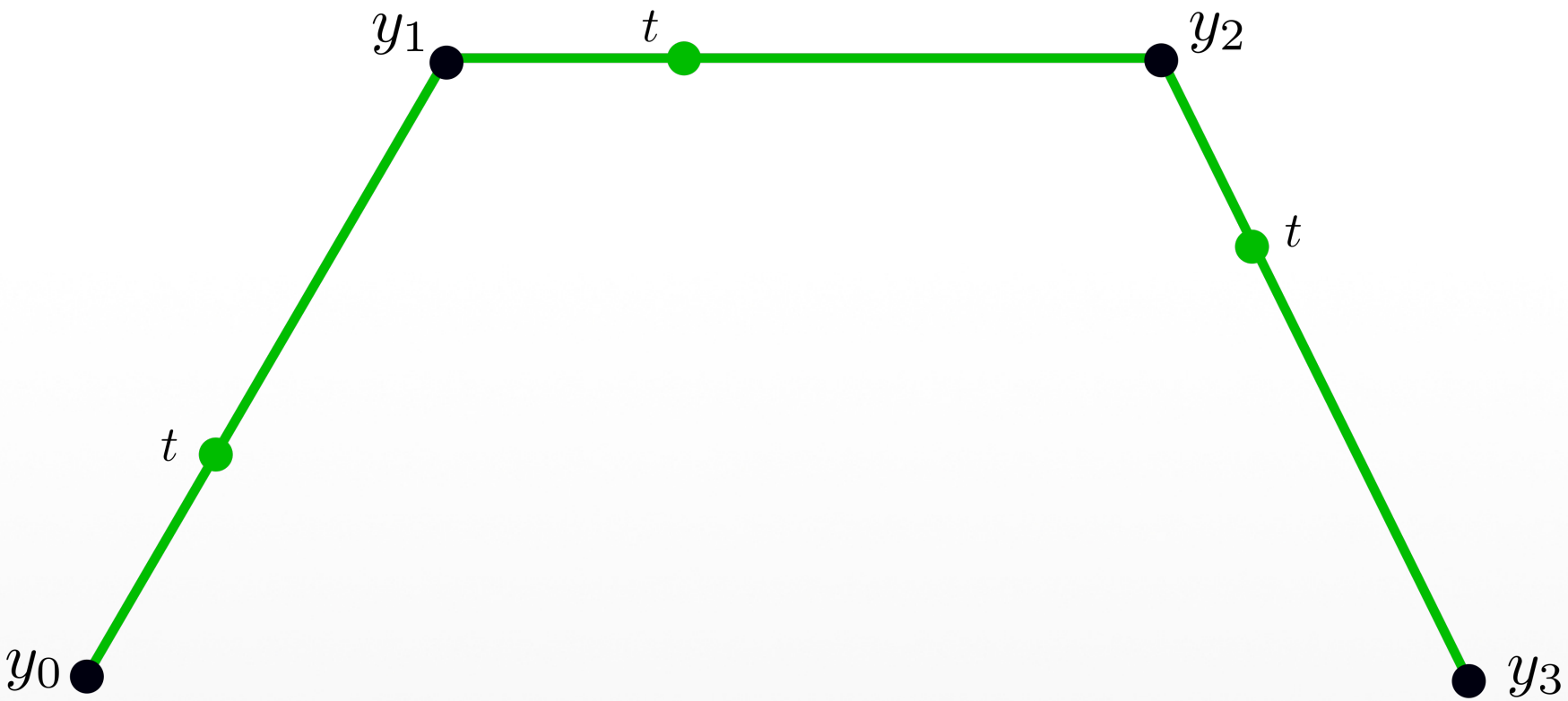
Bézier curves/De Casteljau Algorithm in $\mathbb{R}^n$



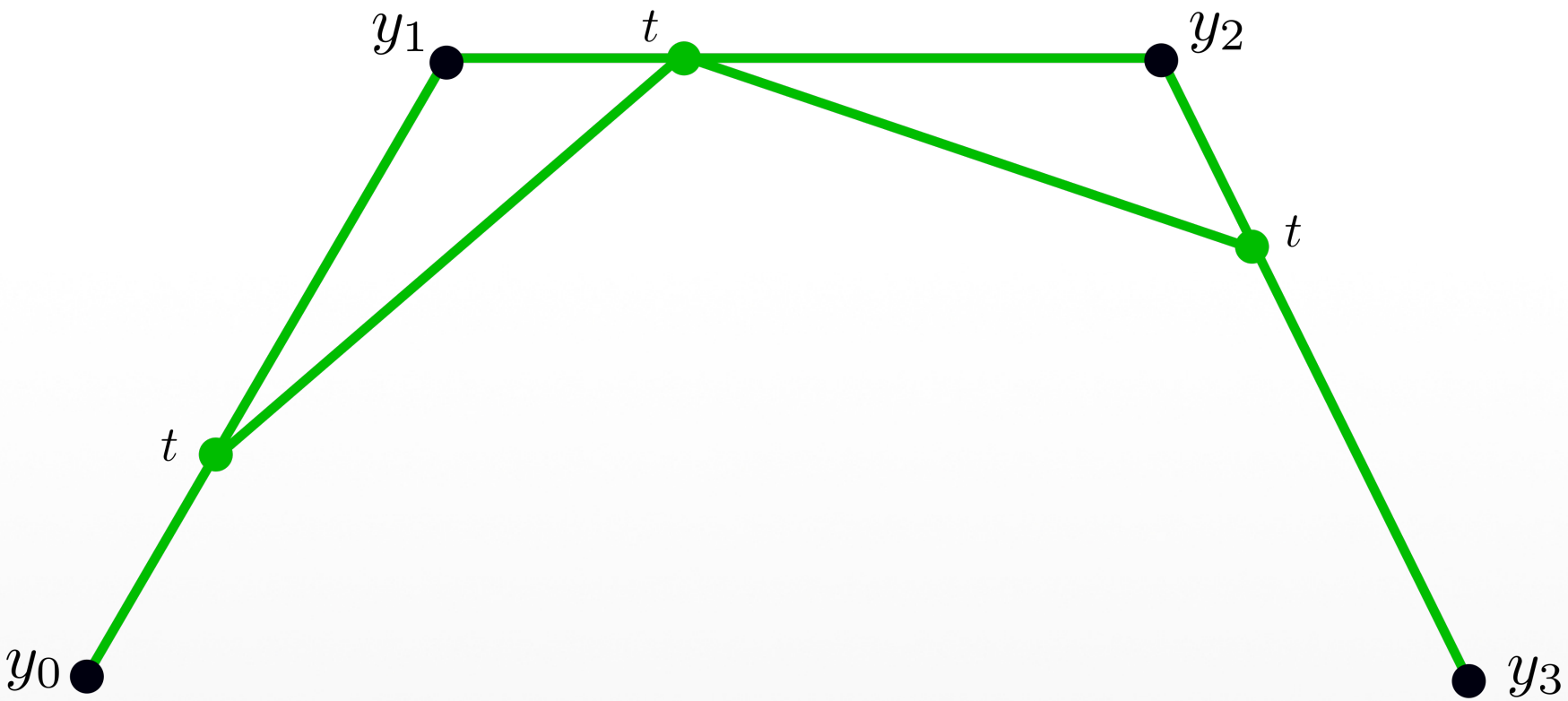
Bézier curves/De Casteljau Algorithm in $\mathbb{R}^n$



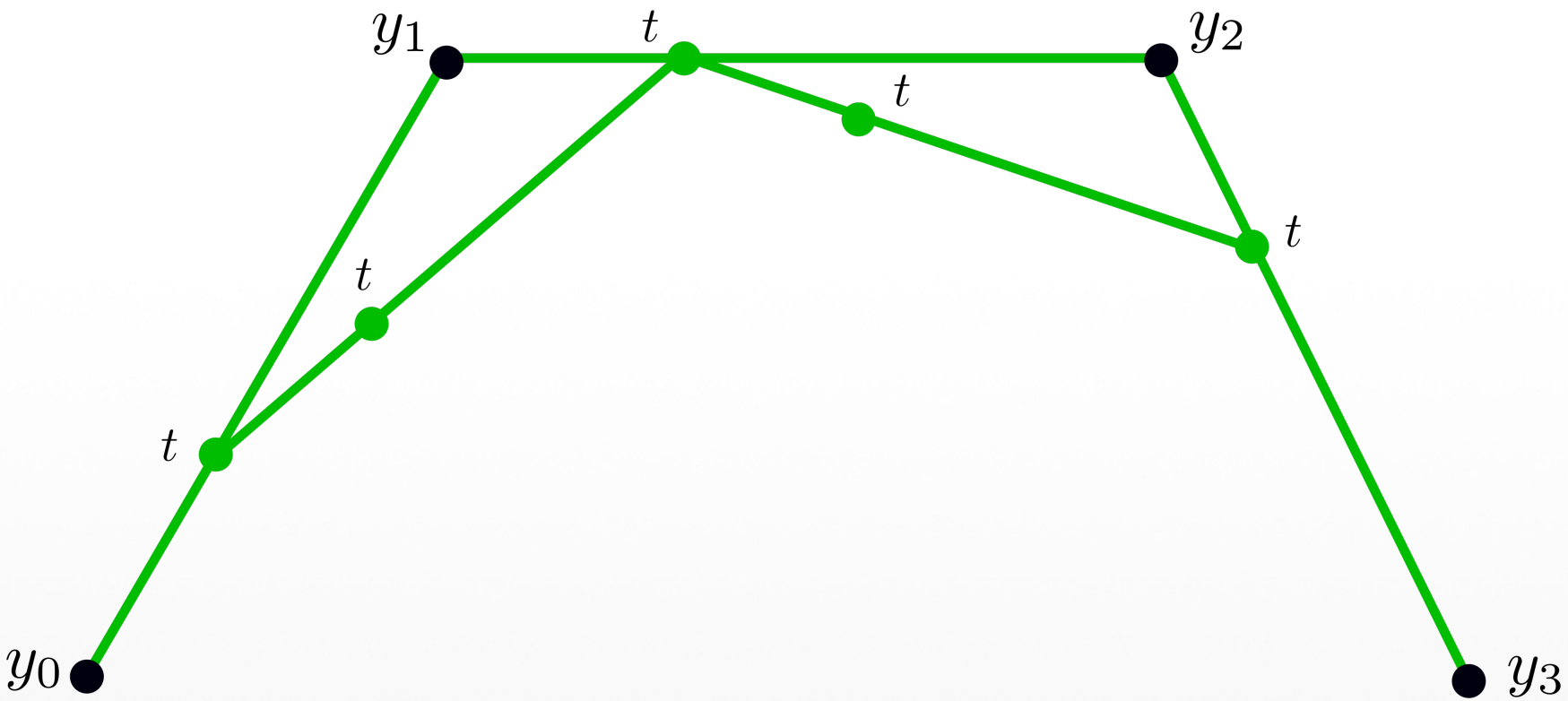
Bézier curves/De Casteljau Algorithm in $\mathbb{R}^n$



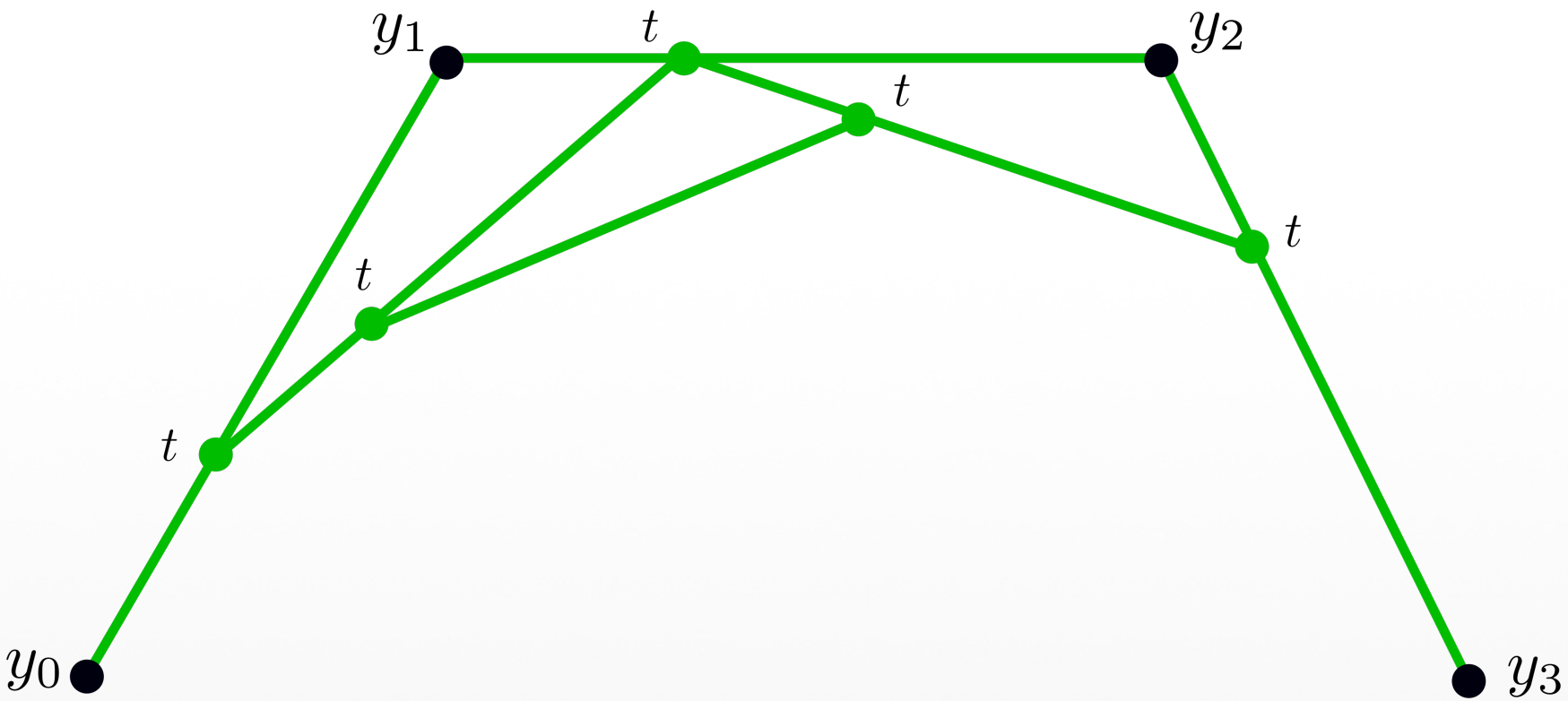
Bézier curves/De Casteljau Algorithm in $\mathbb{R}^n$



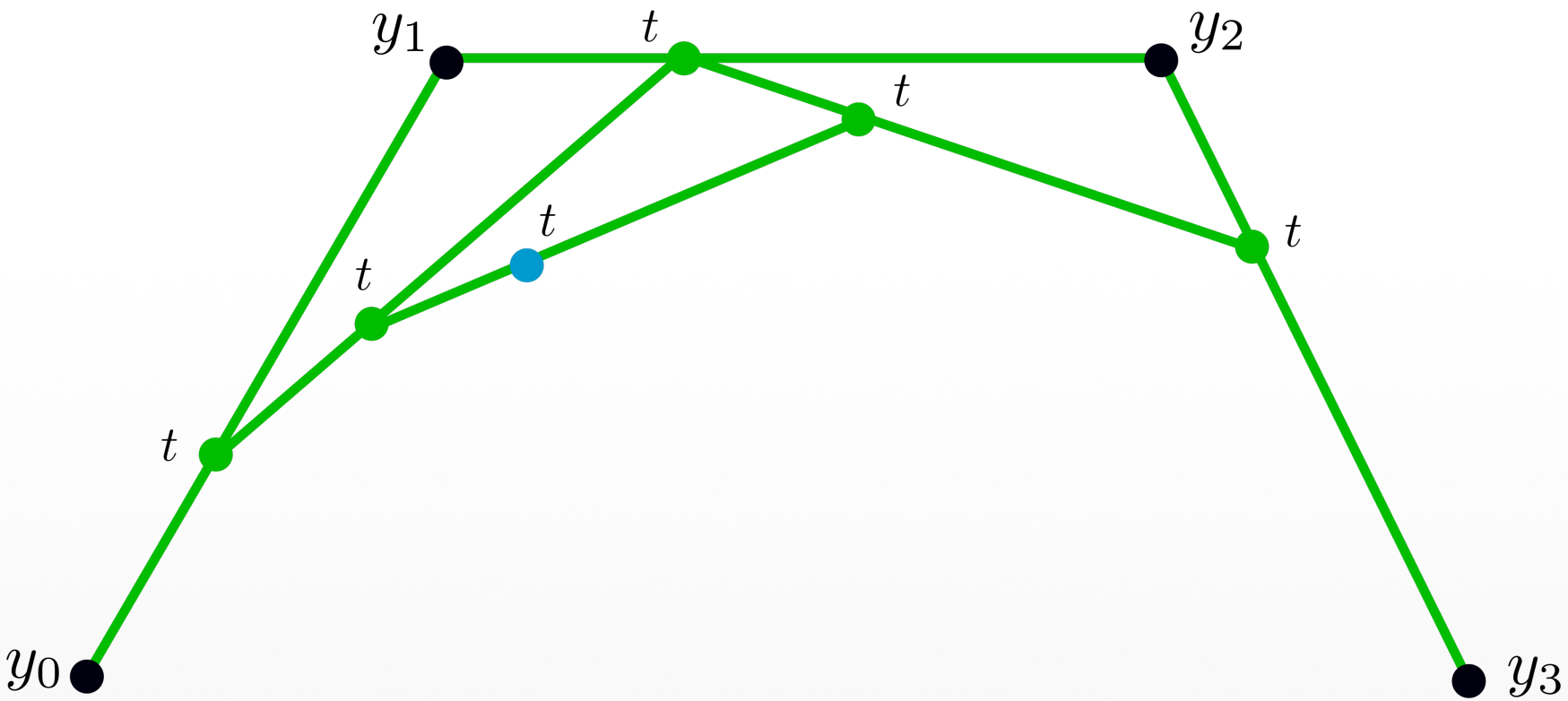
Bézier curves/De Casteljau Algorithm in $\mathbb{R}^n$



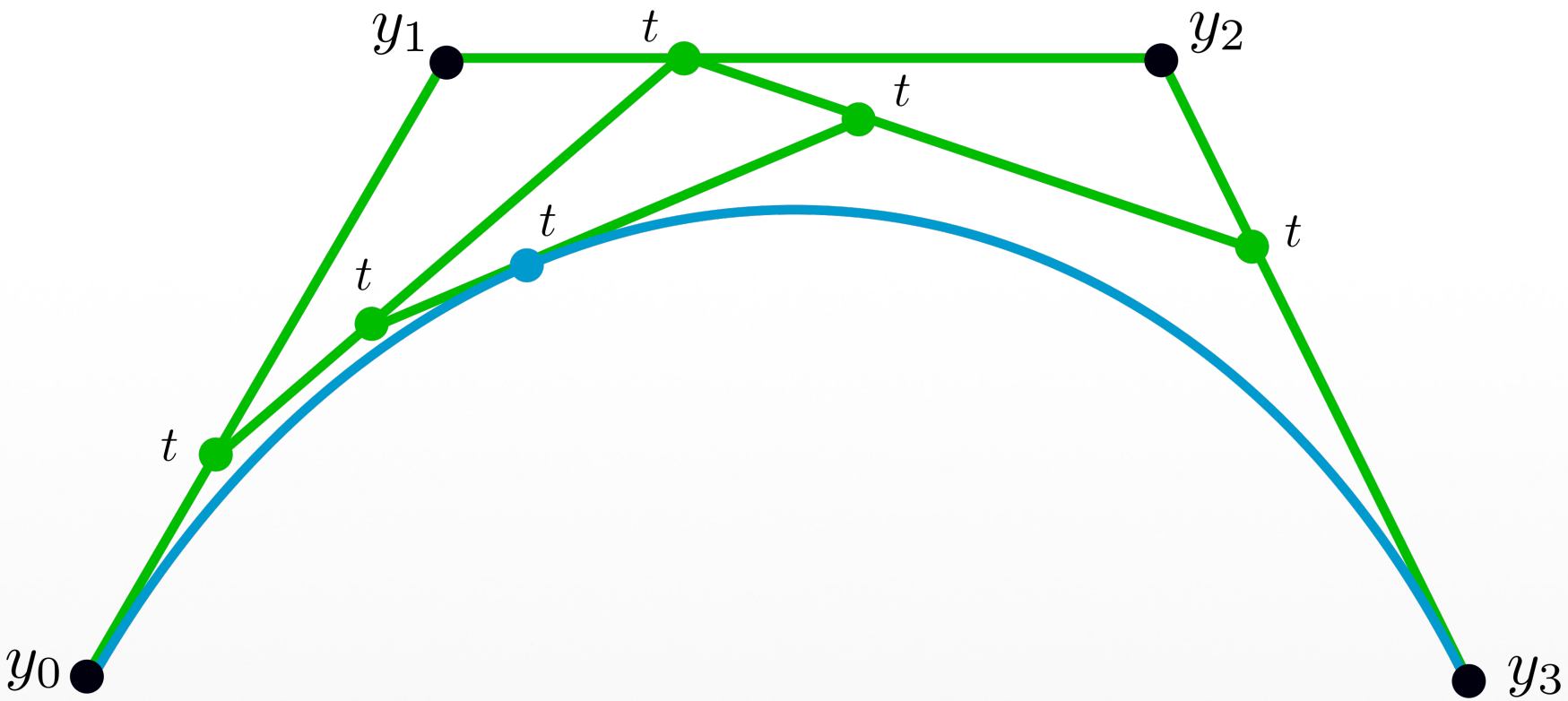
Bézier curves/De Casteljau Algorithm in $\mathbb{R}^n$



Bézier curves/De Casteljau Algorithm in $\mathbb{R}^n$

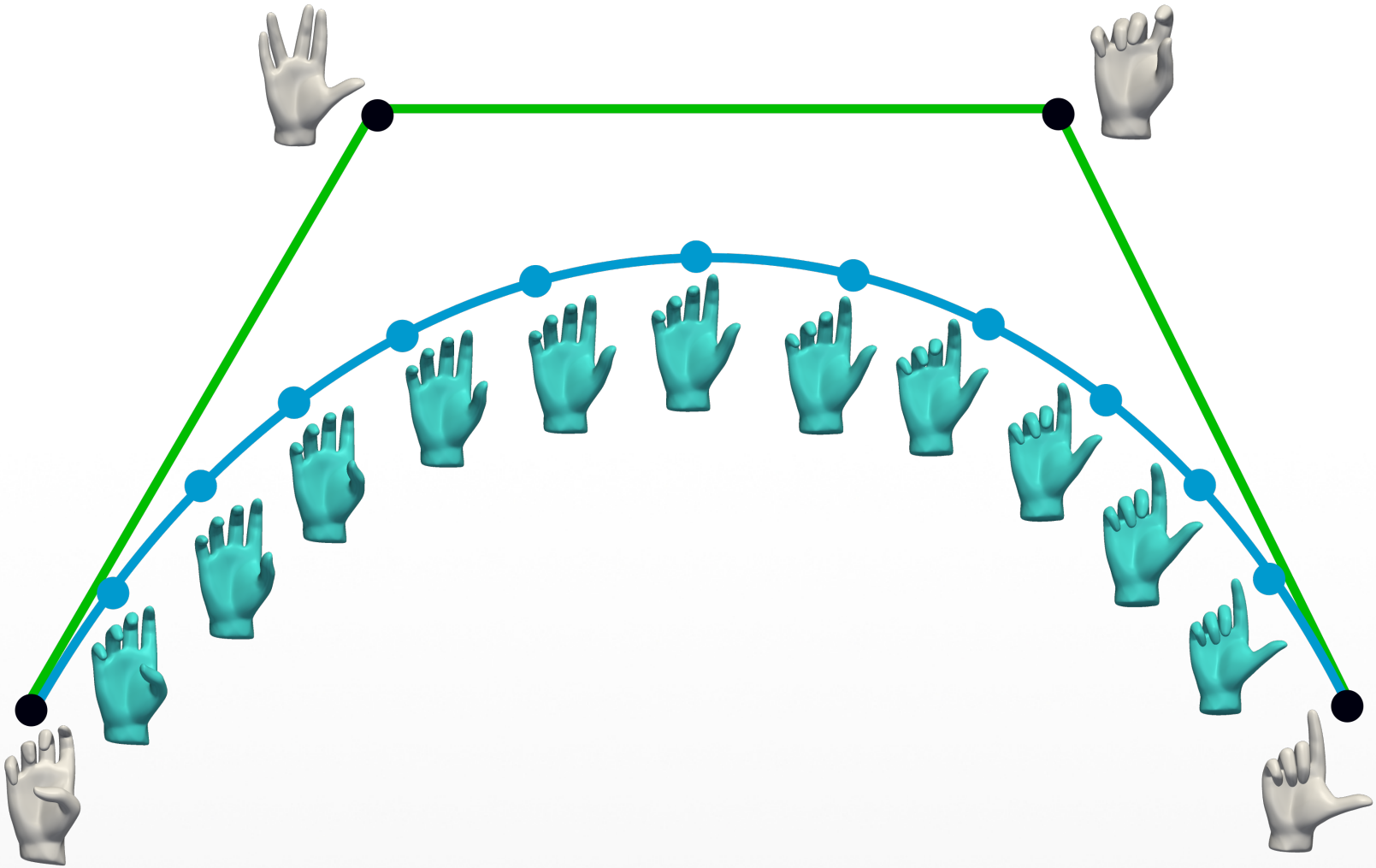


Bézier curves/De Casteljau Algorithm in $\mathbb{R}^n$

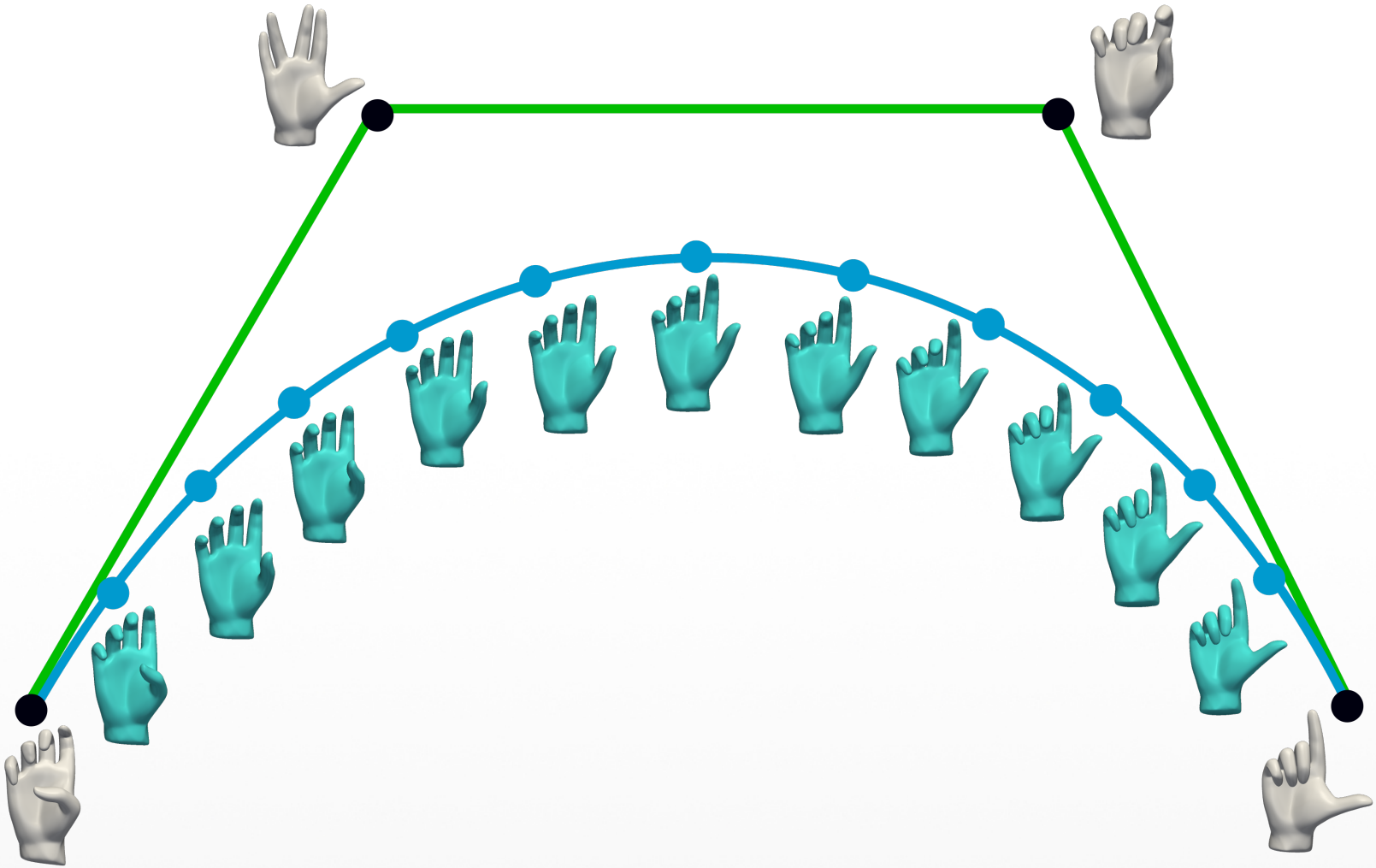


Bézier Curves in Shape Space

Bézier Curves in Shape Space



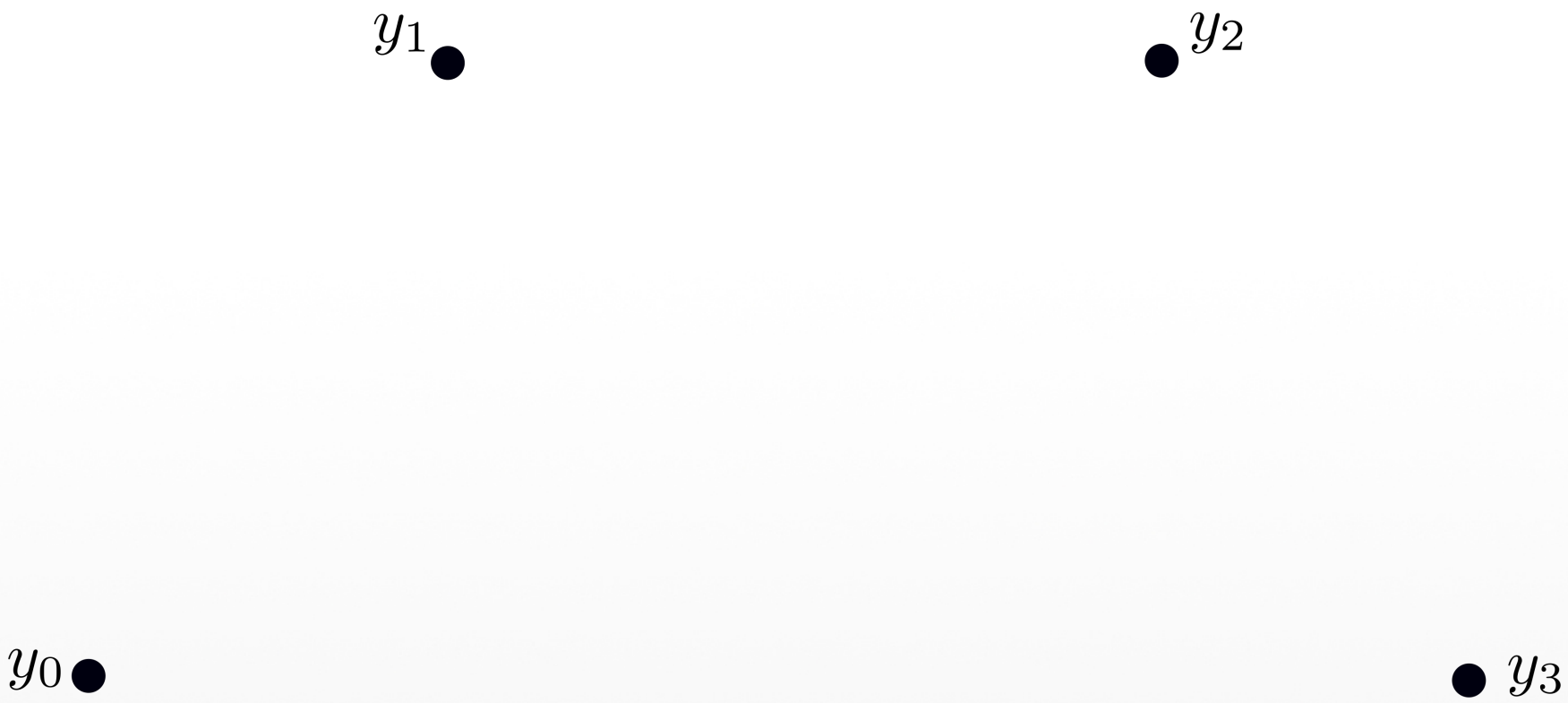
Bézier Curves in Shape Space



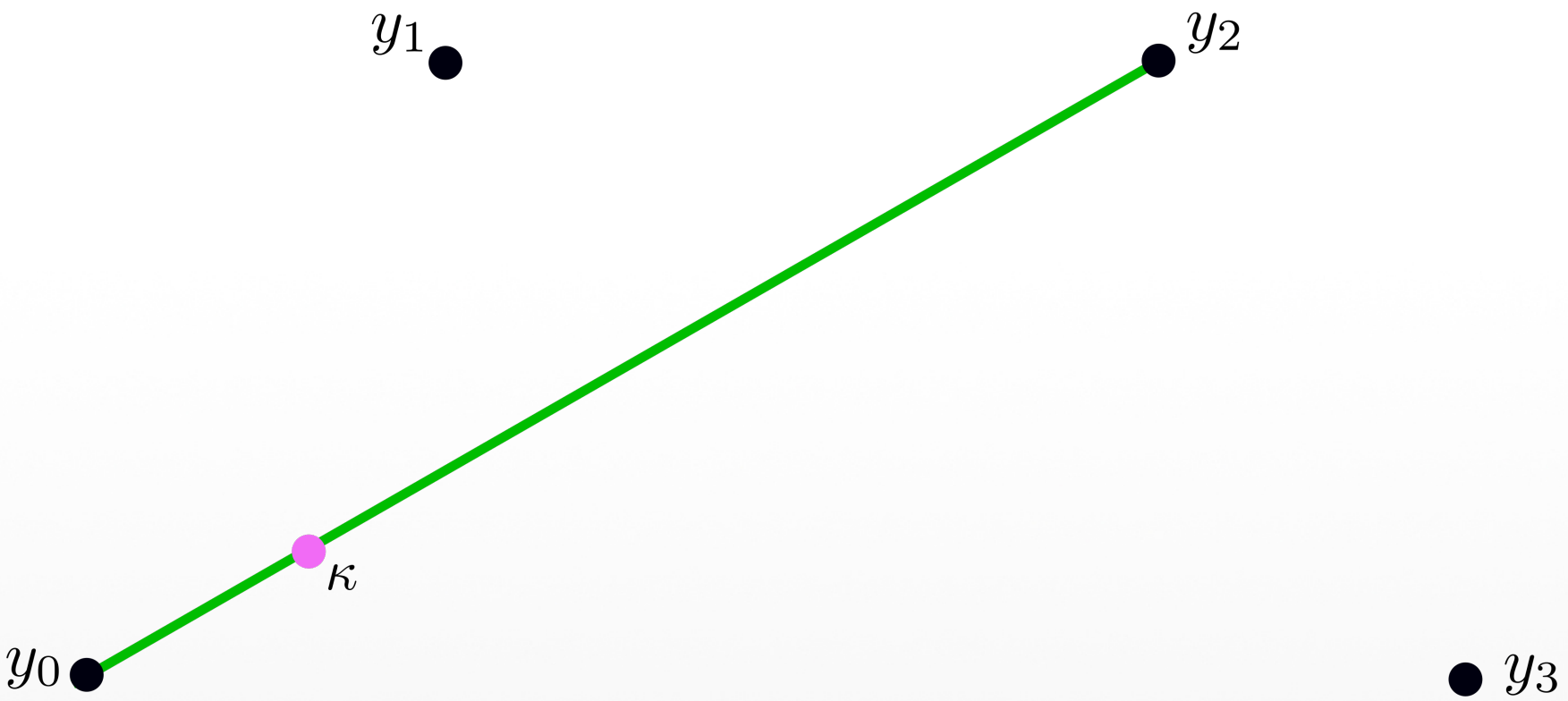
Problems

- Do not interpolate input data
- Evaluation at parameter value depends on all input shapes

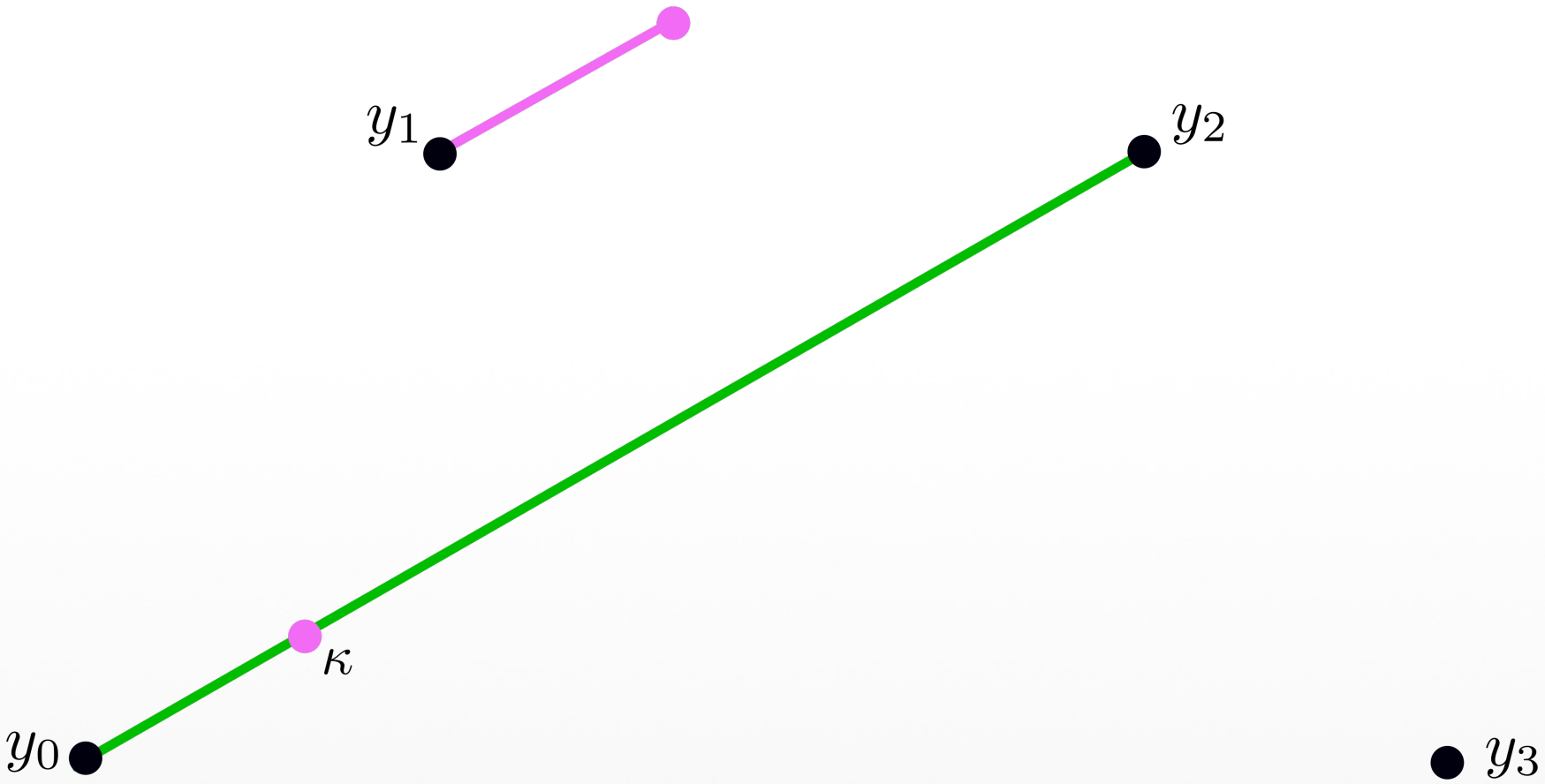
Cardinal Splines in $\mathbb{R}^n$



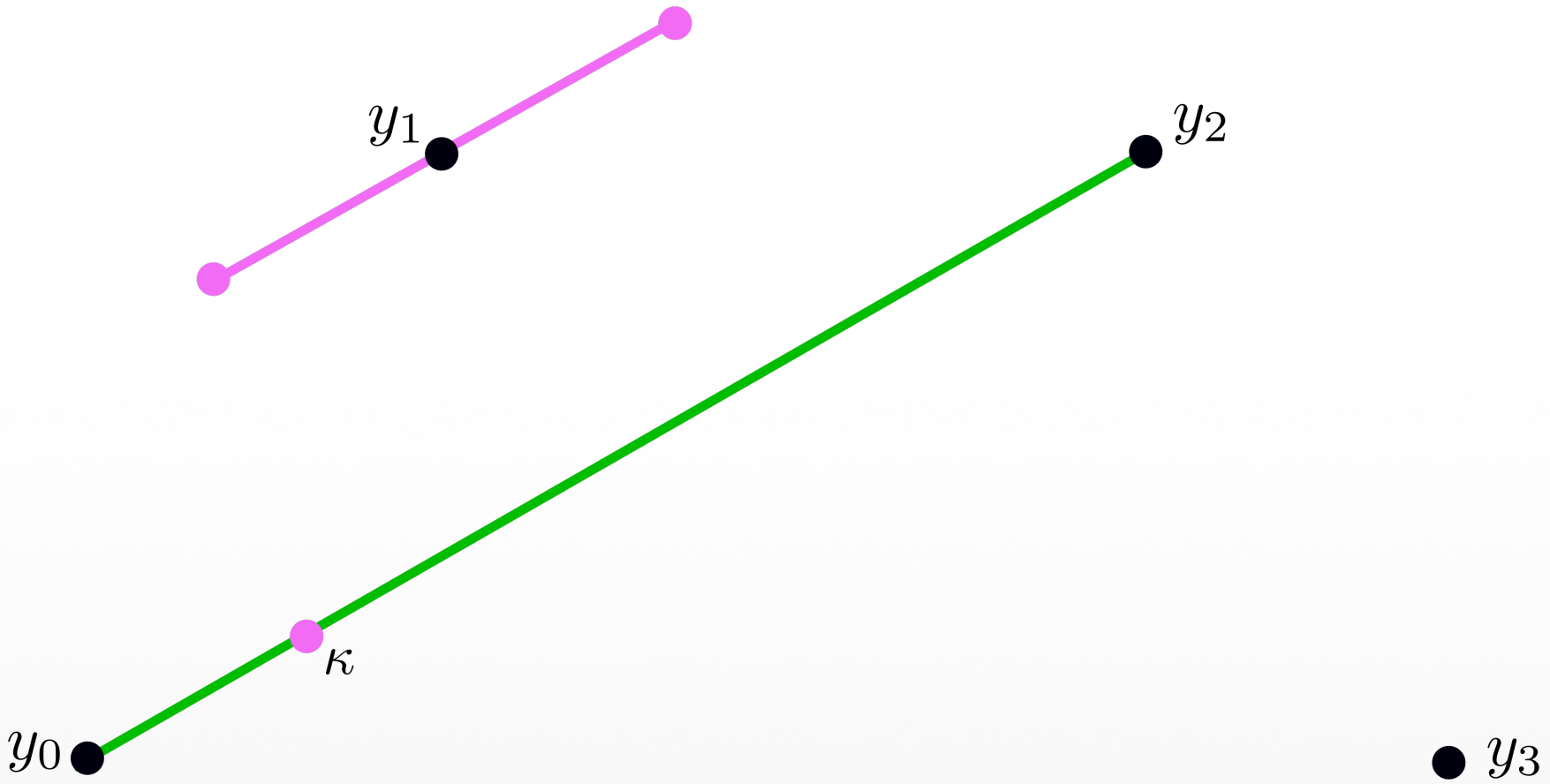
Cardinal Splines in $\mathbb{R}^n$



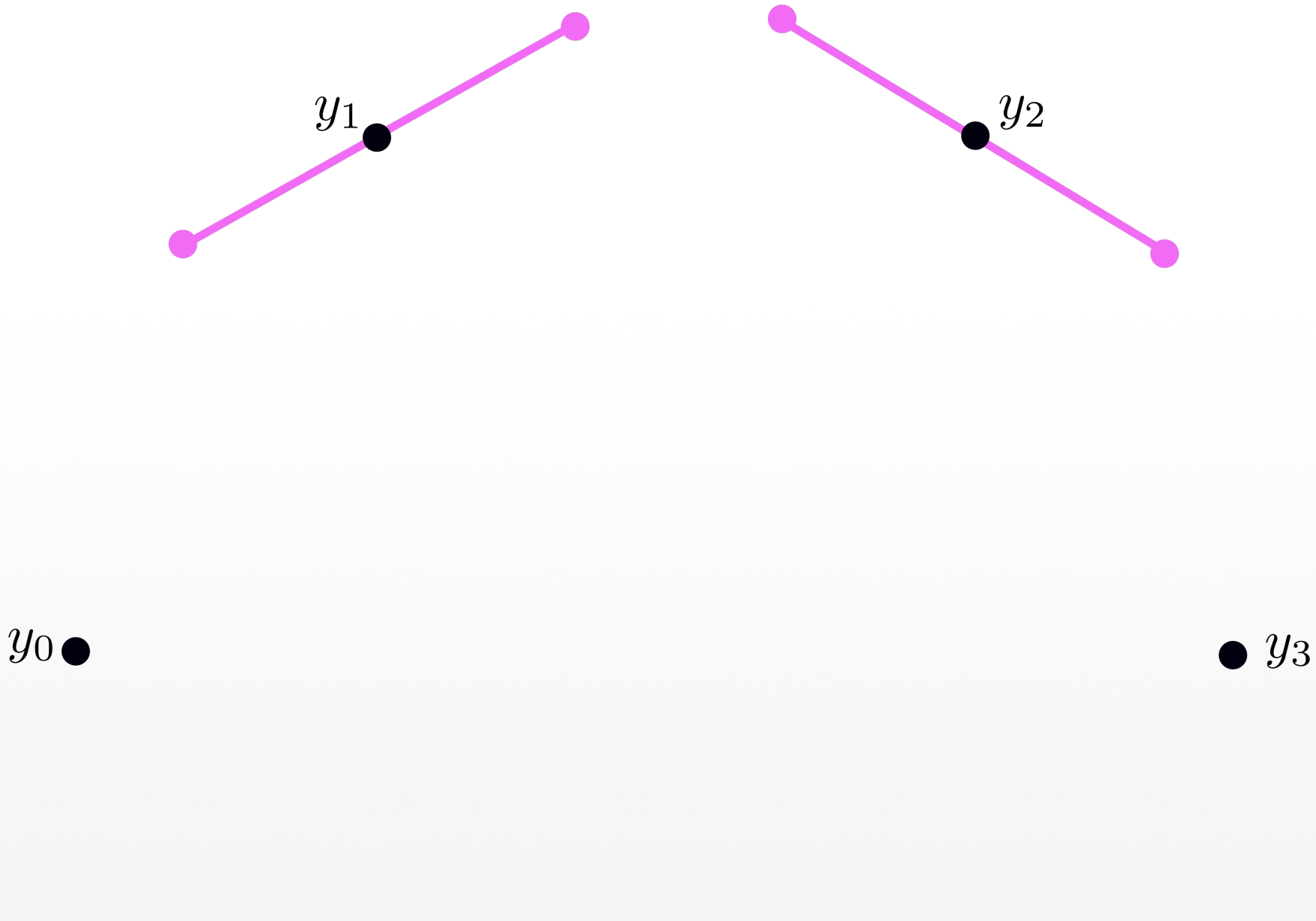
Cardinal Splines in $\mathbb{R}^n$



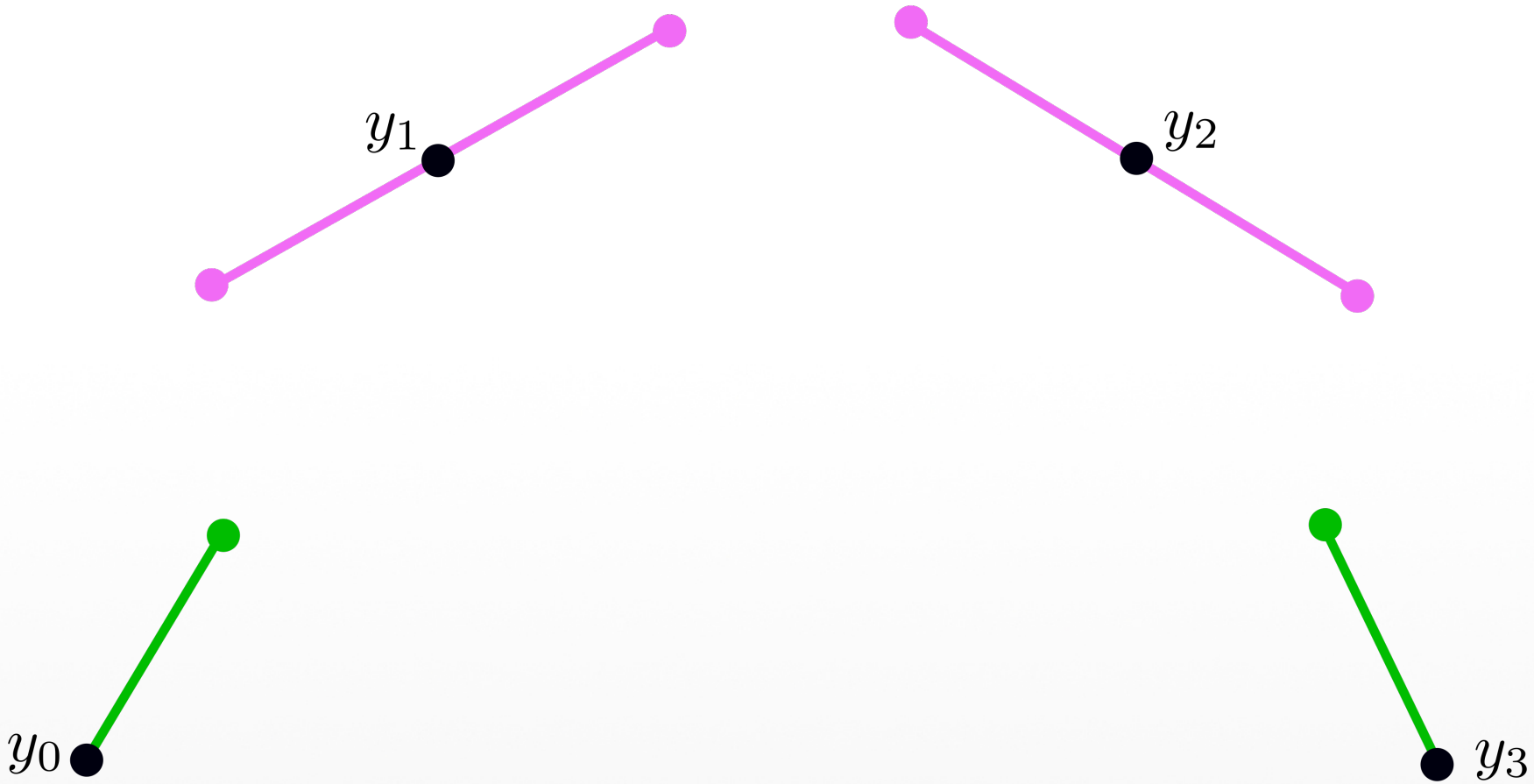
Cardinal Splines in $\mathbb{R}^n$



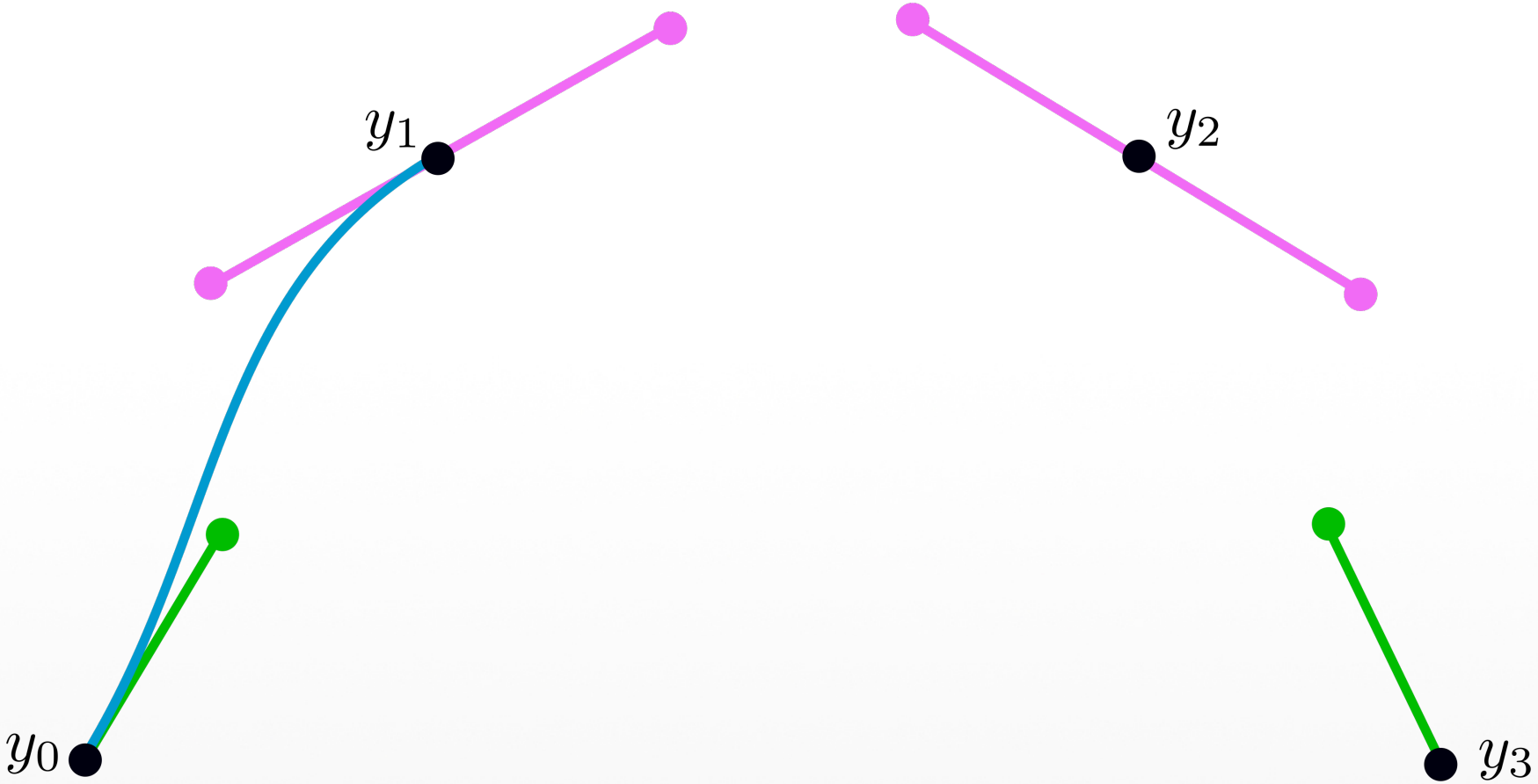
Cardinal Splines in $\mathbb{R}^n$



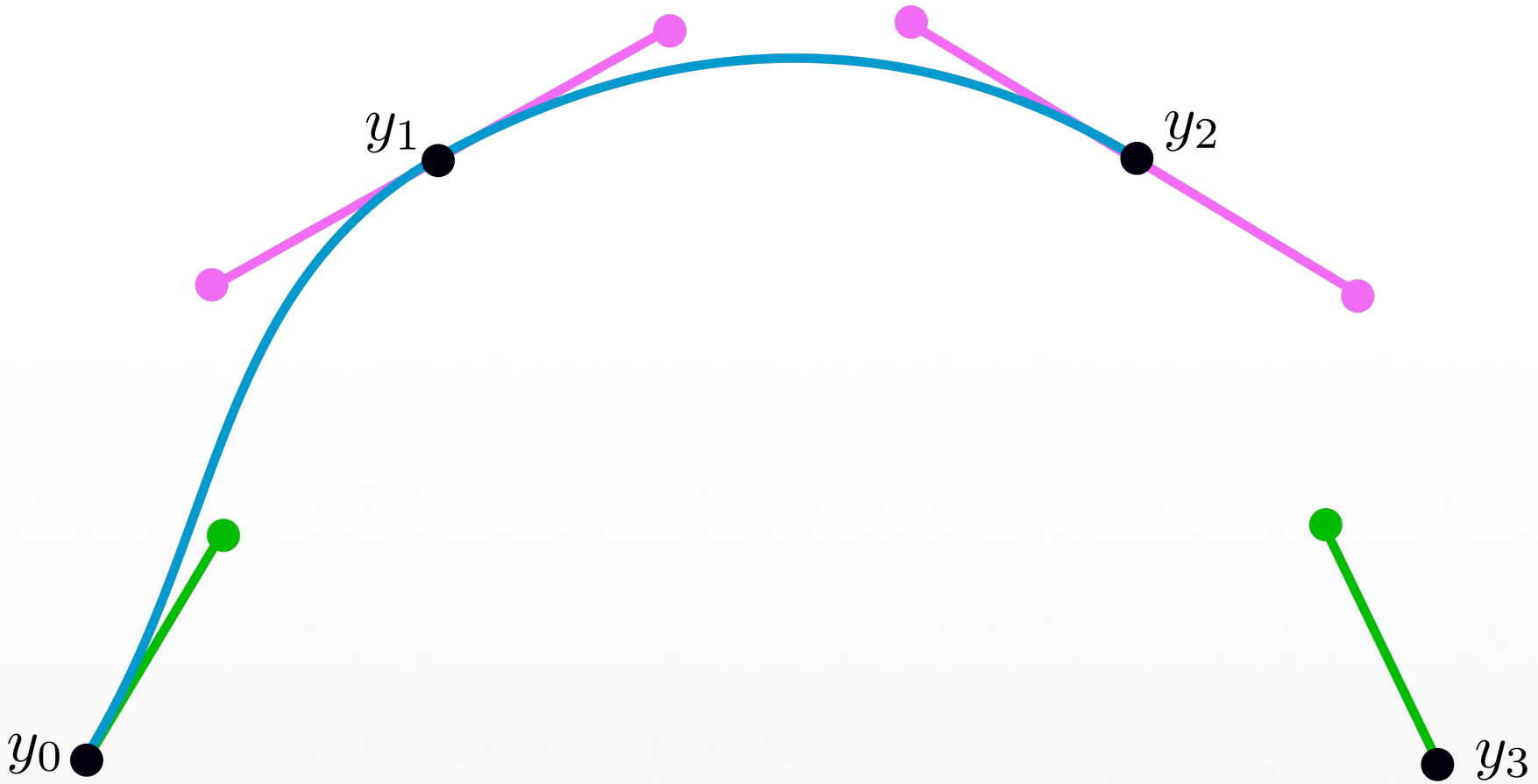
Cardinal Splines in $\mathbb{R}^n$



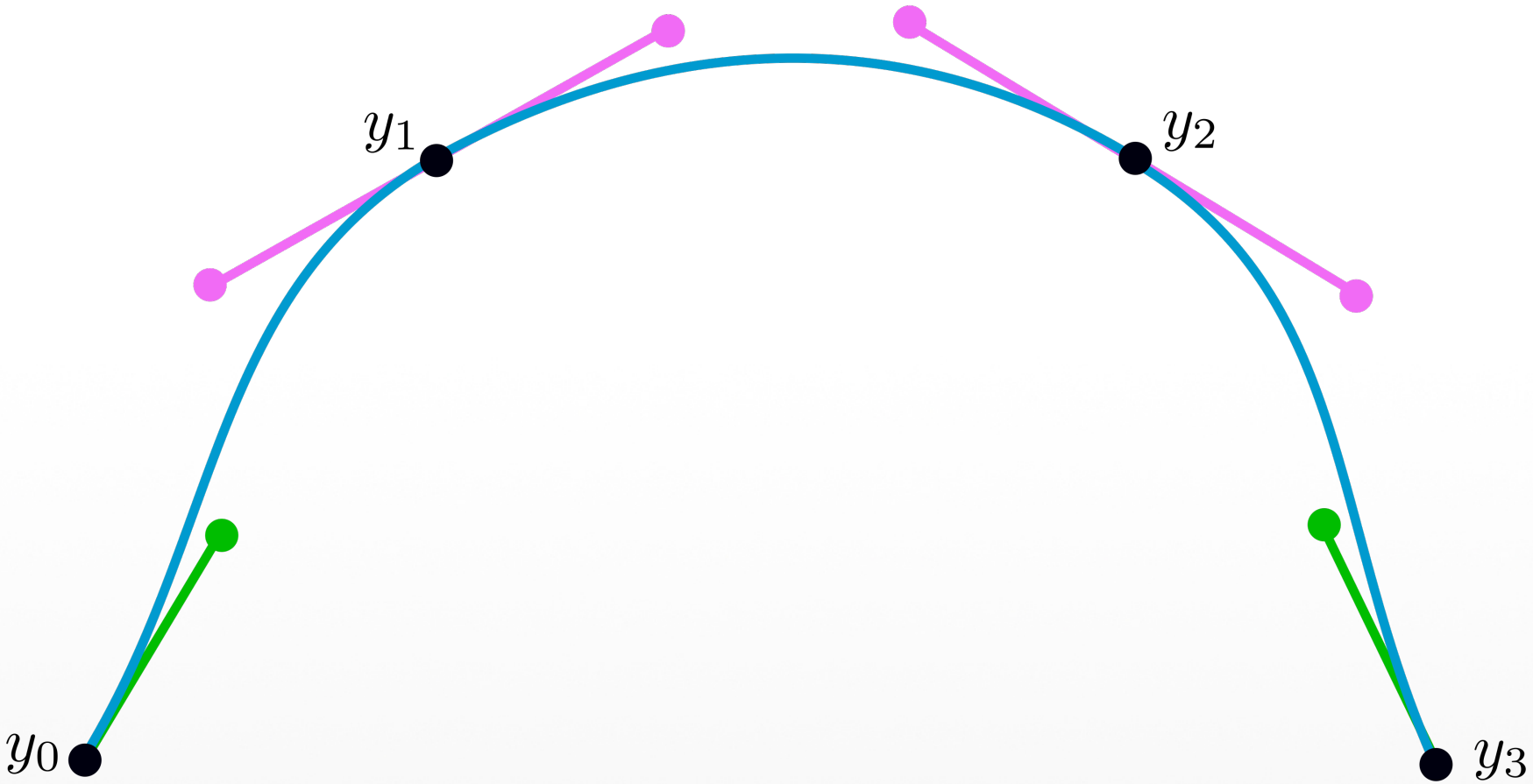
Cardinal Splines in $\mathbb{R}^n$



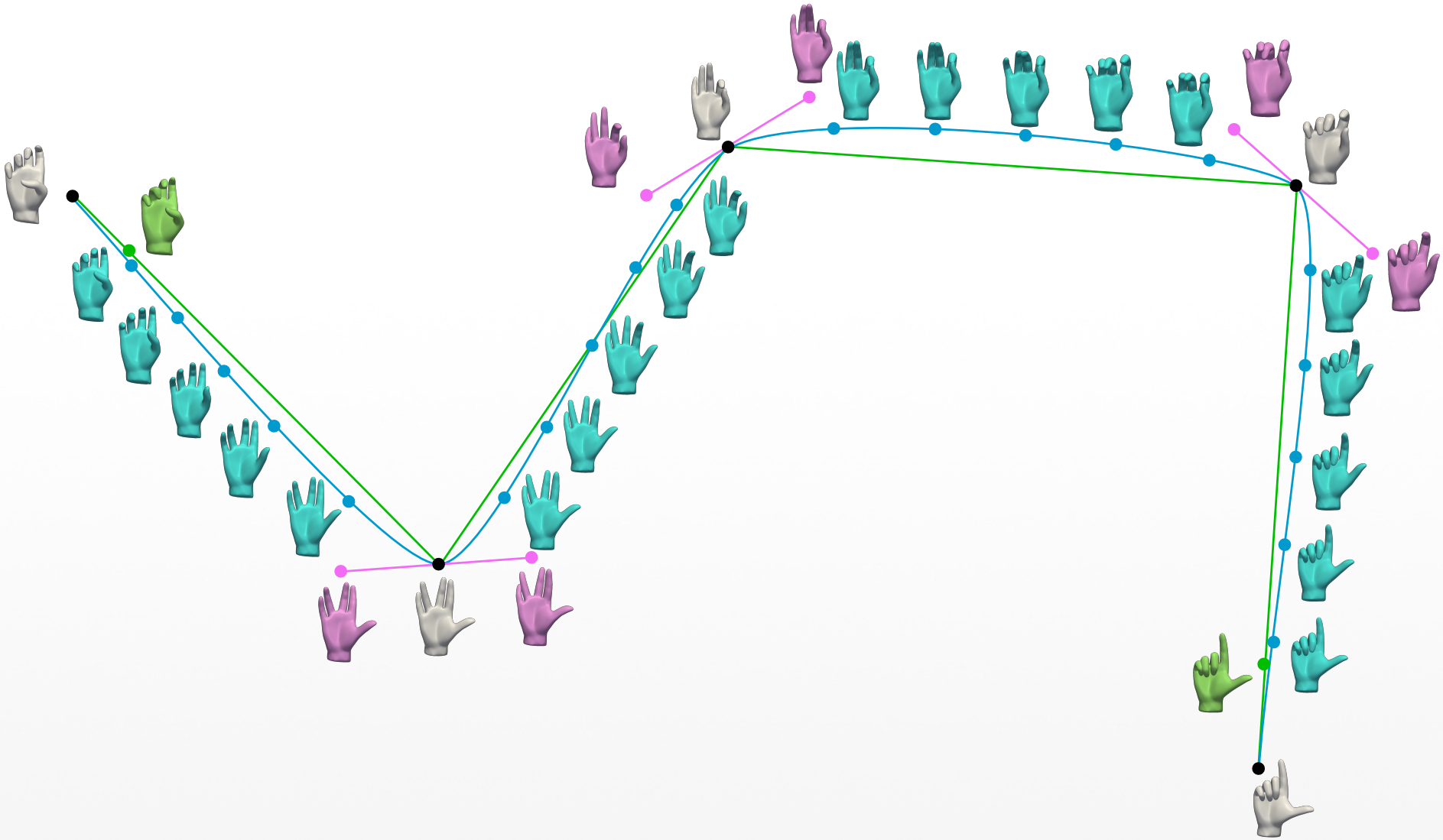
Cardinal Splines in $\mathbb{R}^n$



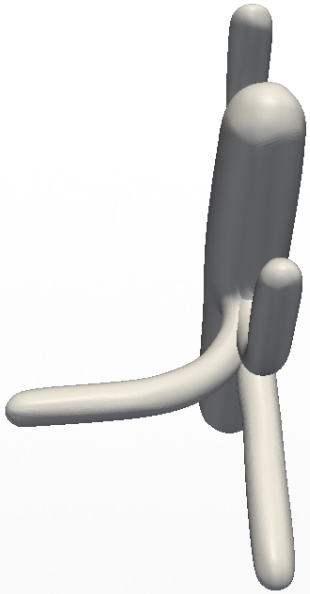
Cardinal Splines in $\mathbb{R}^n$



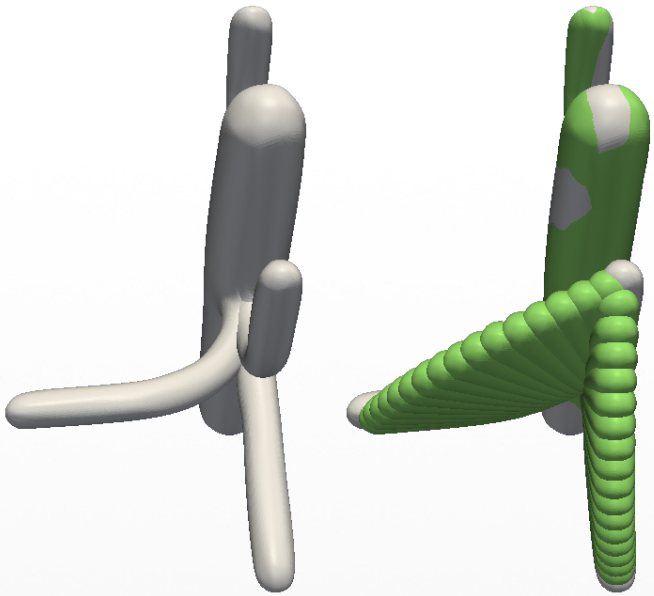
Cardinal Splines in Shape Space



Tension parameter

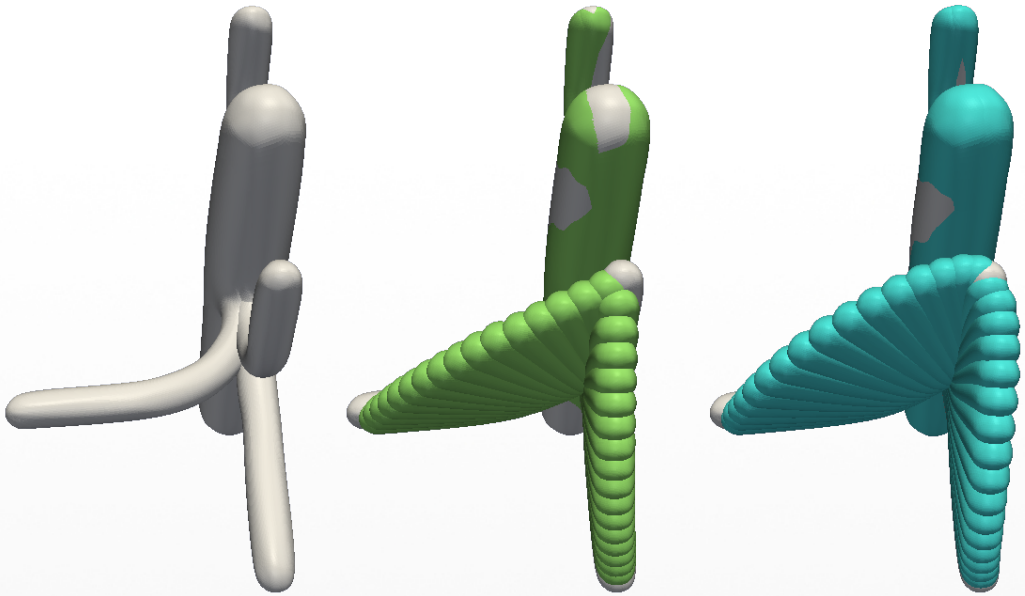


Tension parameter



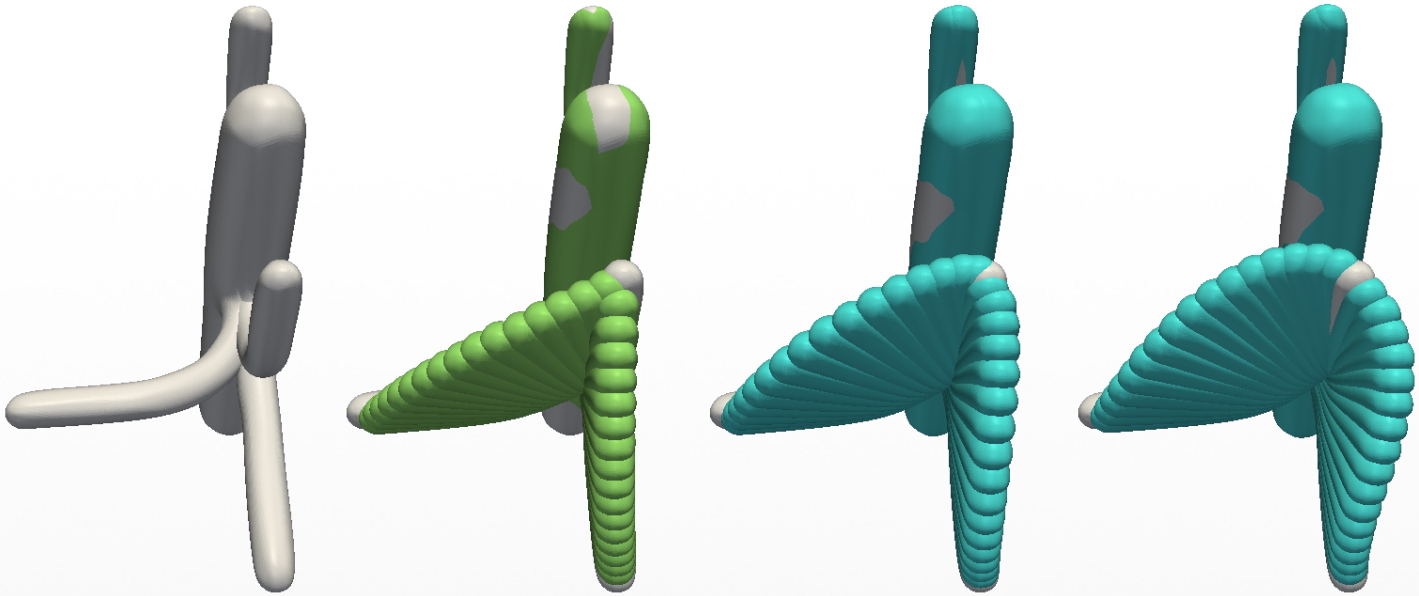
$\kappa = 0$

Tension parameter



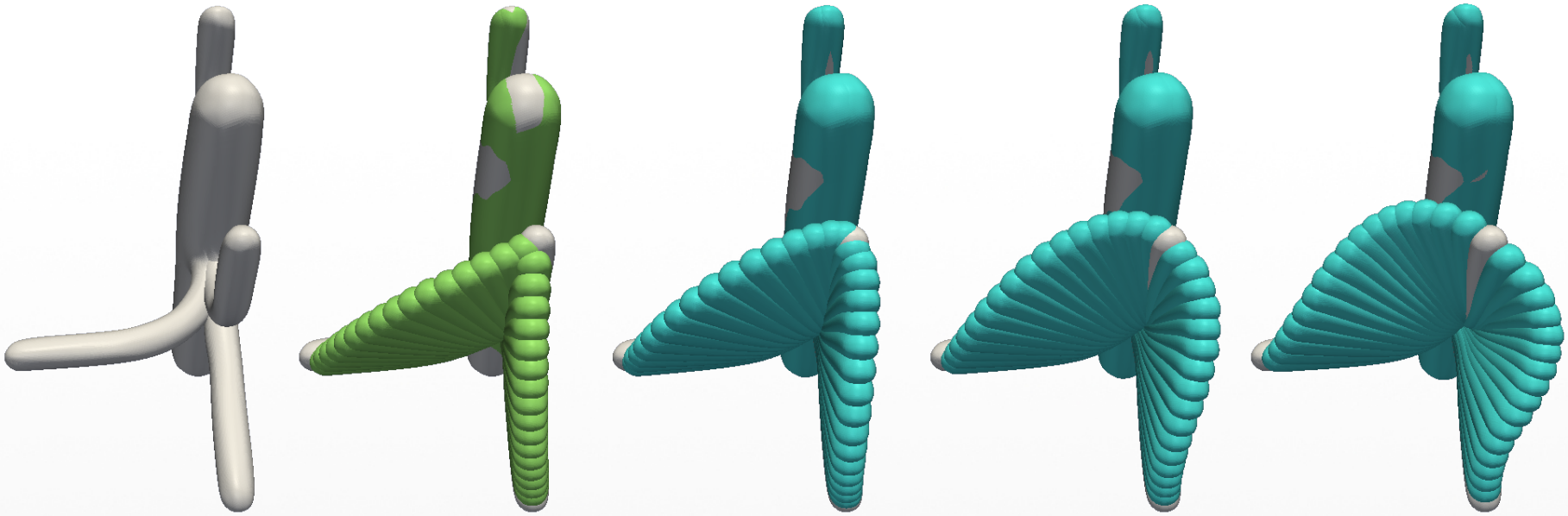
$\kappa = 0$ $\kappa = 0.5$

Tension parameter



$\kappa = 0$ $\kappa = 0.5$ $\kappa = 1$

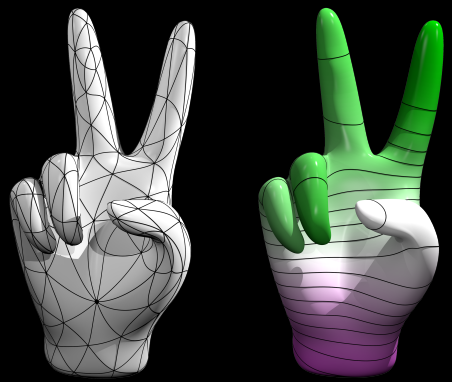
Tension parameter



$\kappa = 0$ $\kappa = 0.5$ $\kappa = 1$ $\kappa = 1.5$

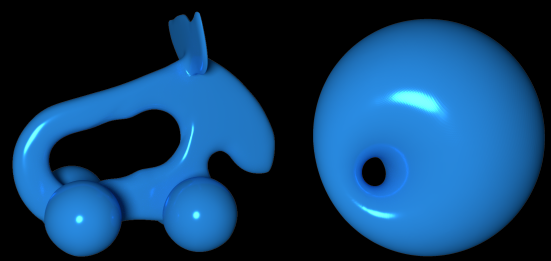
Overview

Isogeometric
Subdivision Method



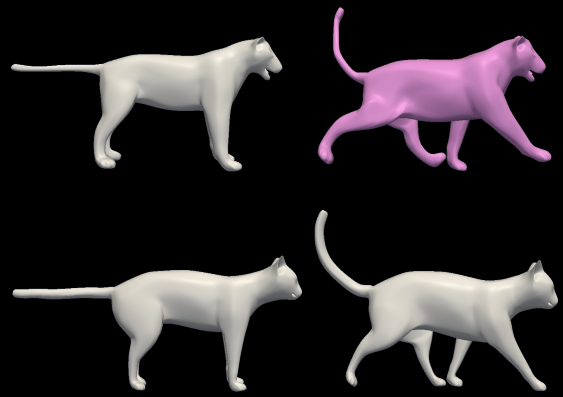
⇒ Robust and efficient
numerical integration

Parametric
Gradient Flows



⇒ Higher order
Time and Space
Discretization

Shape Space



⇒ Bézier curves
and splines