Reduced-Order Shape Optimization Using Offset Surfaces

Przemyslaw Musialski[1]

Thomas Auzinger[1]

Michael Birsak[1]

Michael Wimmer[1]

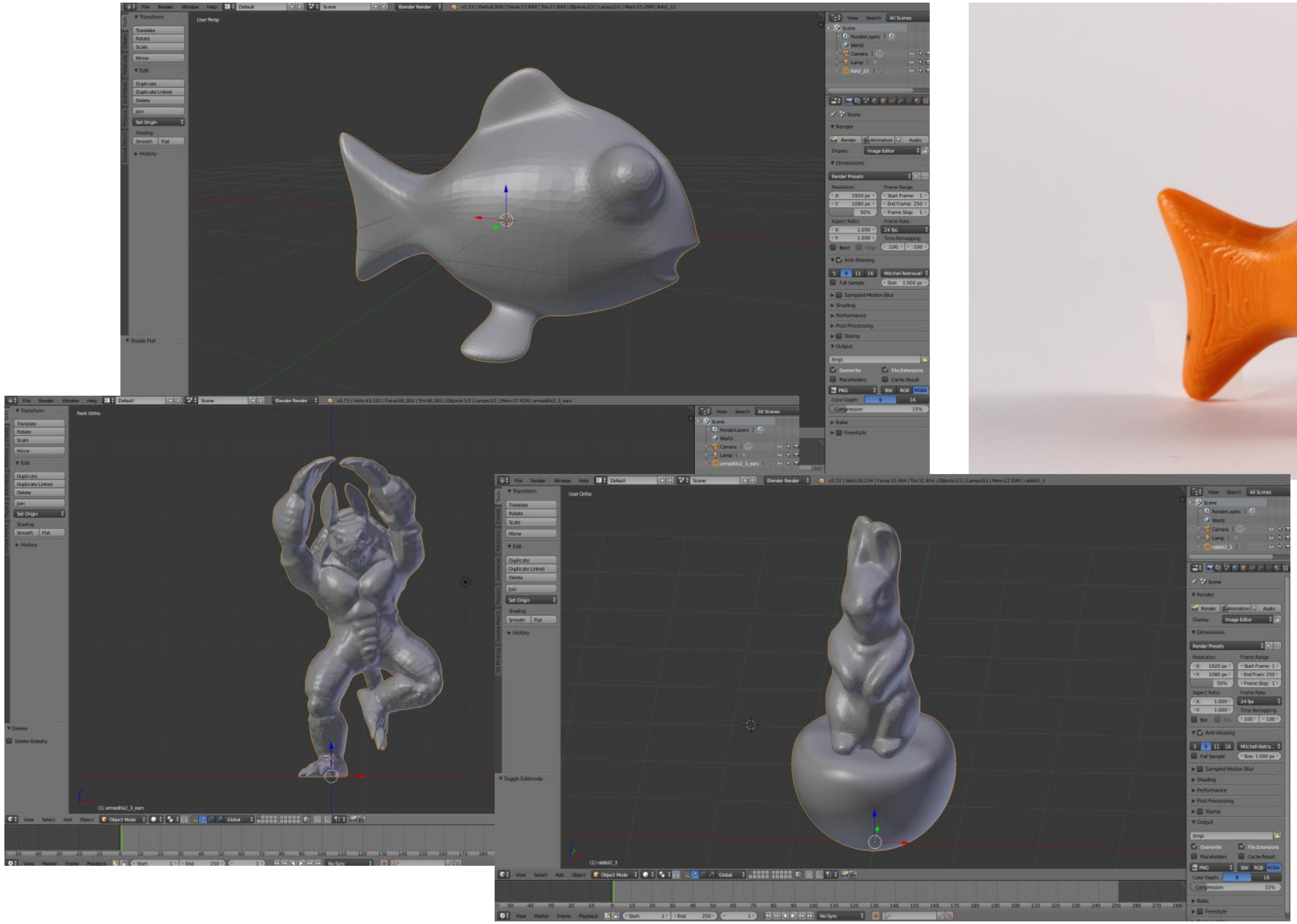
Leif Kobbelt[2]

[1] TU Wien

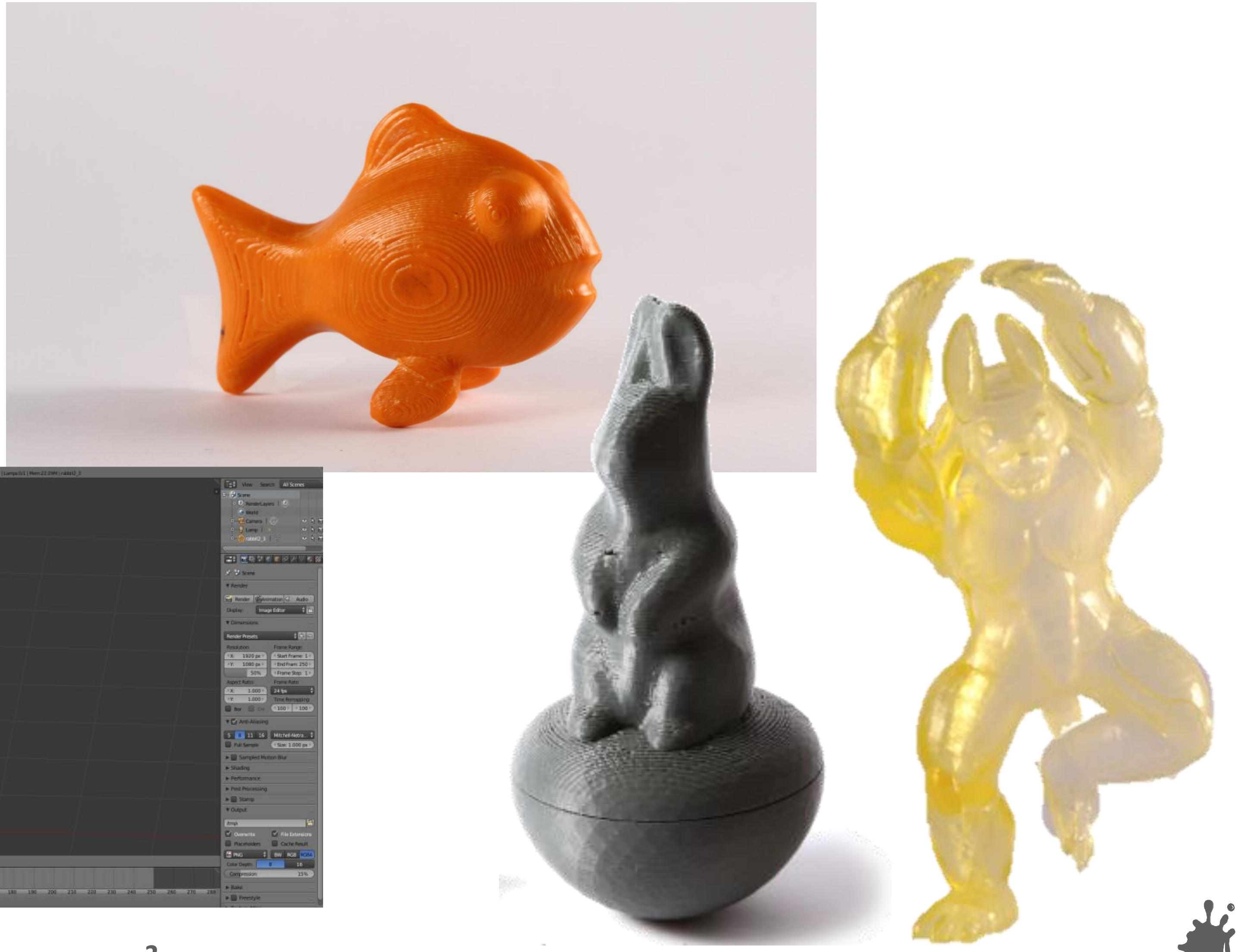
[2] RWTH Aachen

Motivation

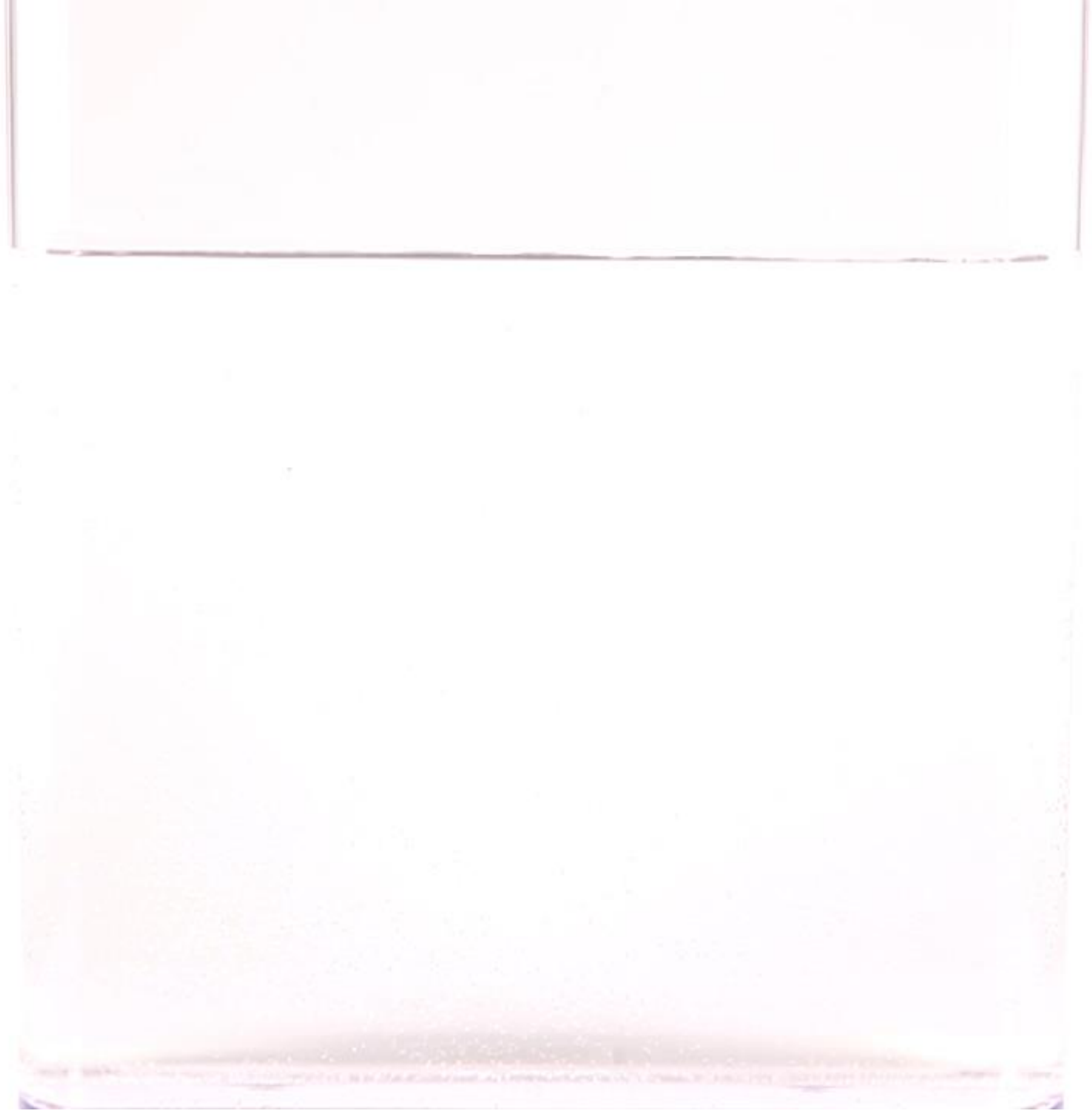
3D Modeling…

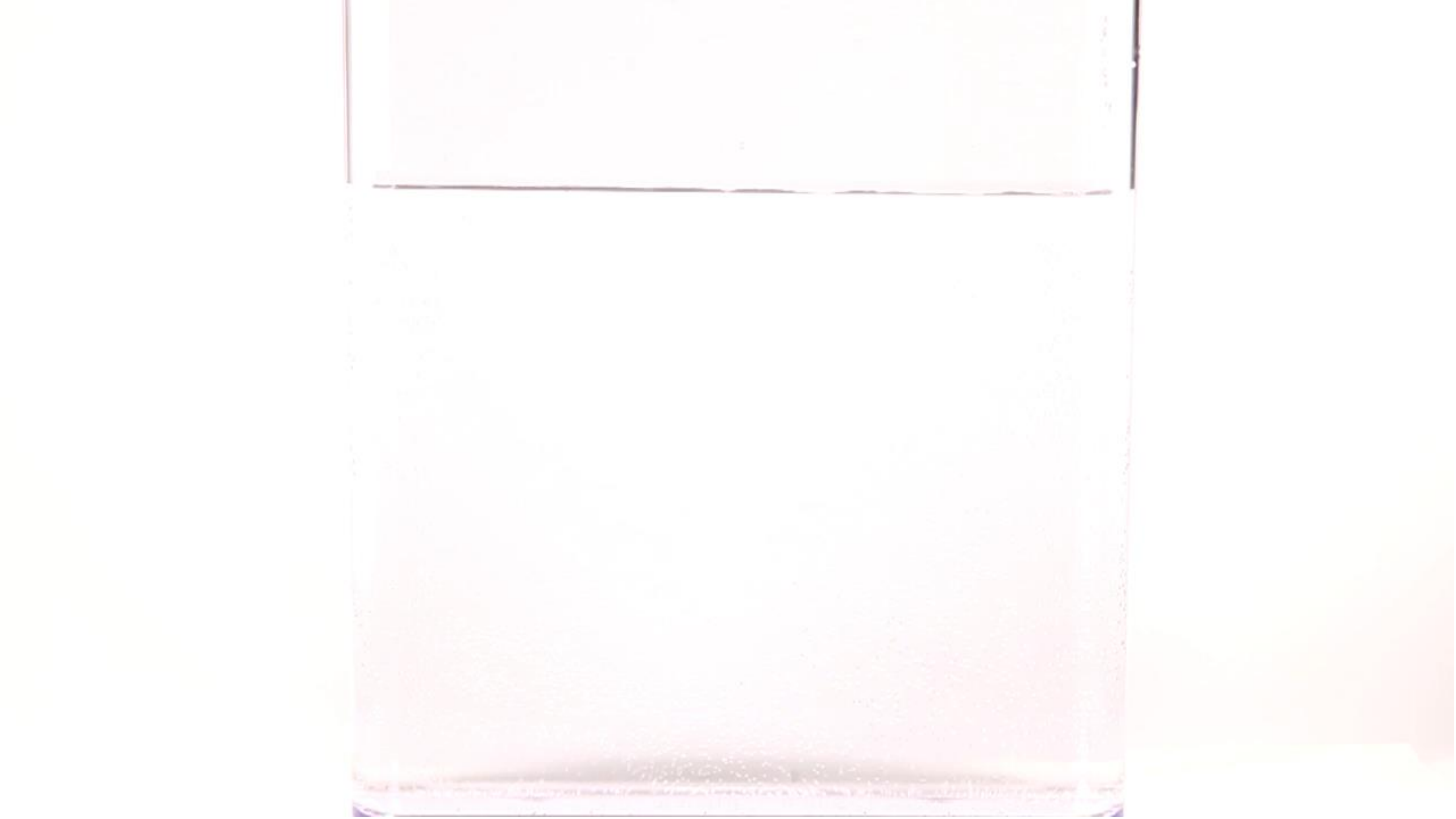


3D Printing…



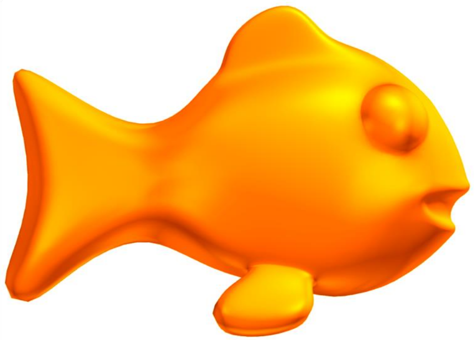
Motivation





Goals

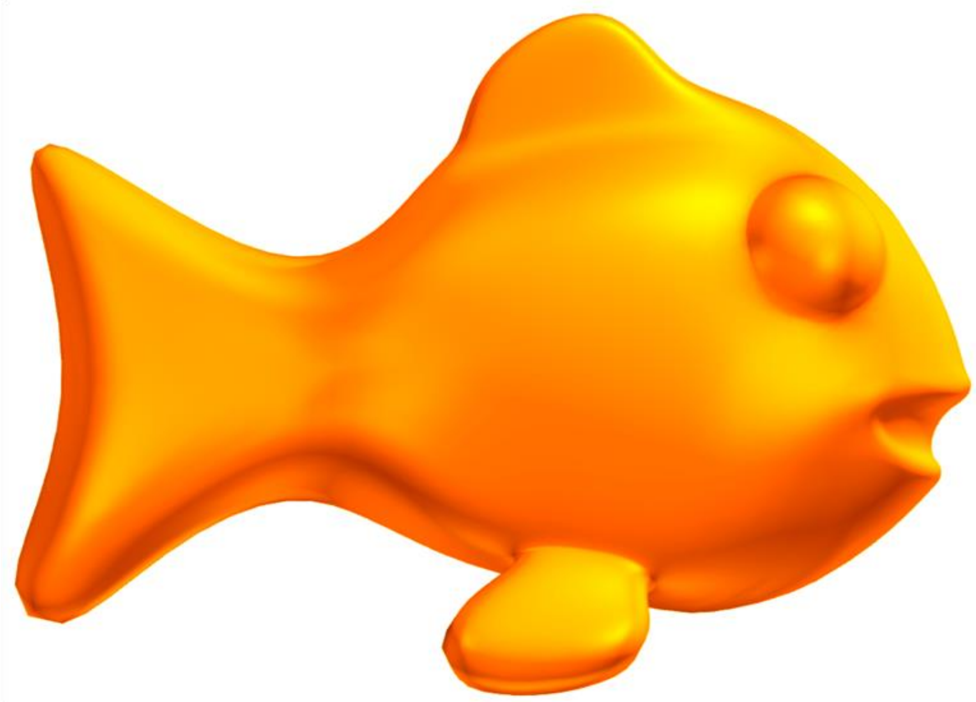
1. Optimize the shape to fulfill the desired goals



Goals

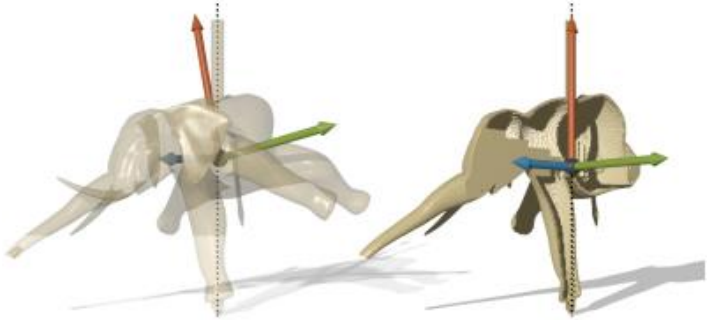
1. Optimize the shape to fulfill the desired goals

2. Keep the input shape deformation minimal

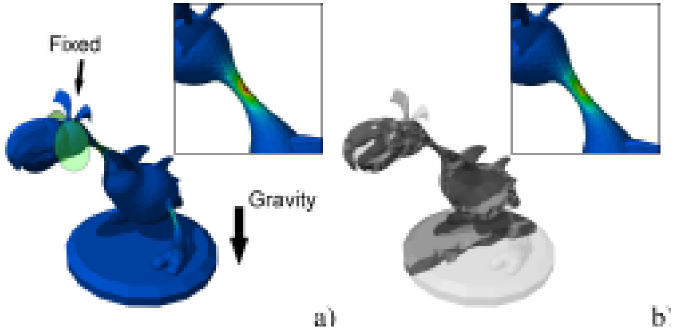


Related Work

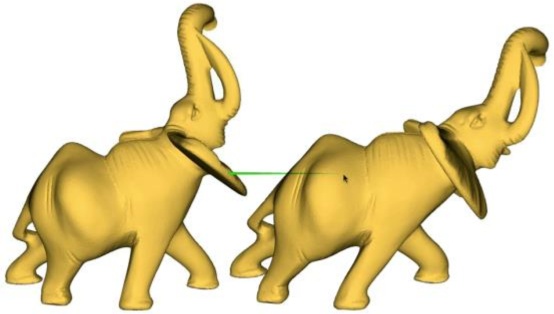
- Optimization of Mass Properties
  - [Prevost et al. 2013]
  - [Baecher et al. 2014]



- Structural Optimization
  - [Stava et al. 2012]
  - [Lu et al. 2014]

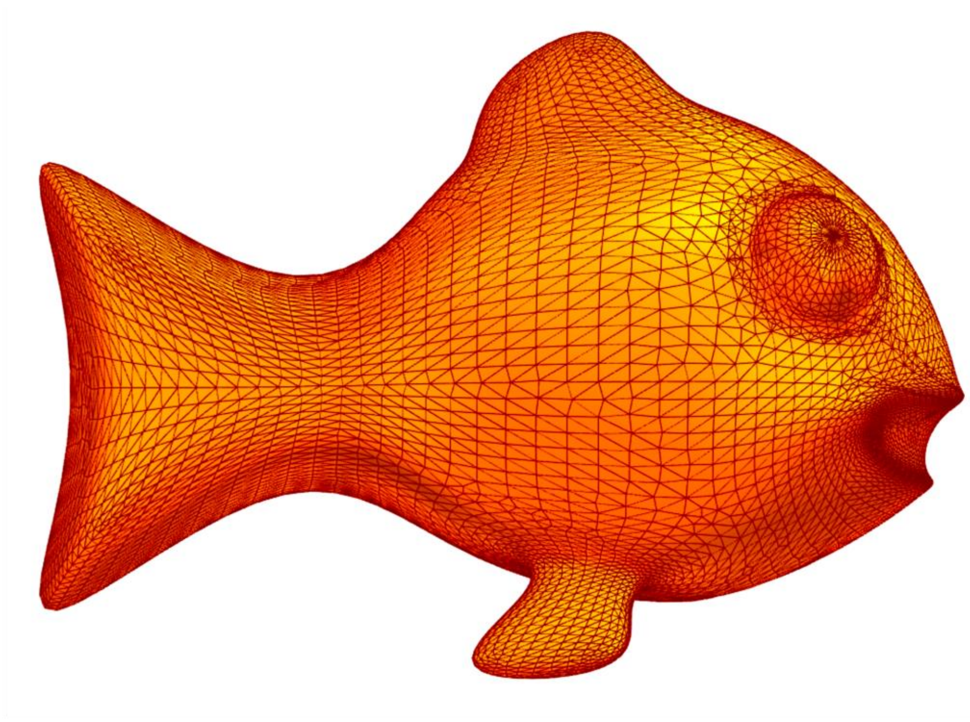


- Reduced Order Models
  - [Pentland and Williams 1989]
  - [von Tycowitch et al. 2013]



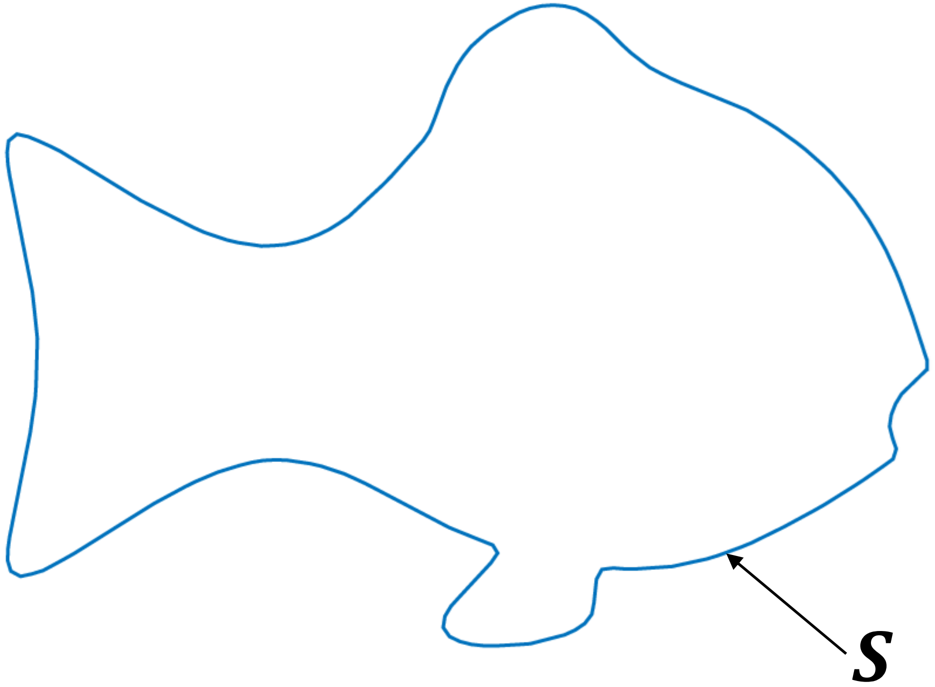
Shape Representation

Input and Output



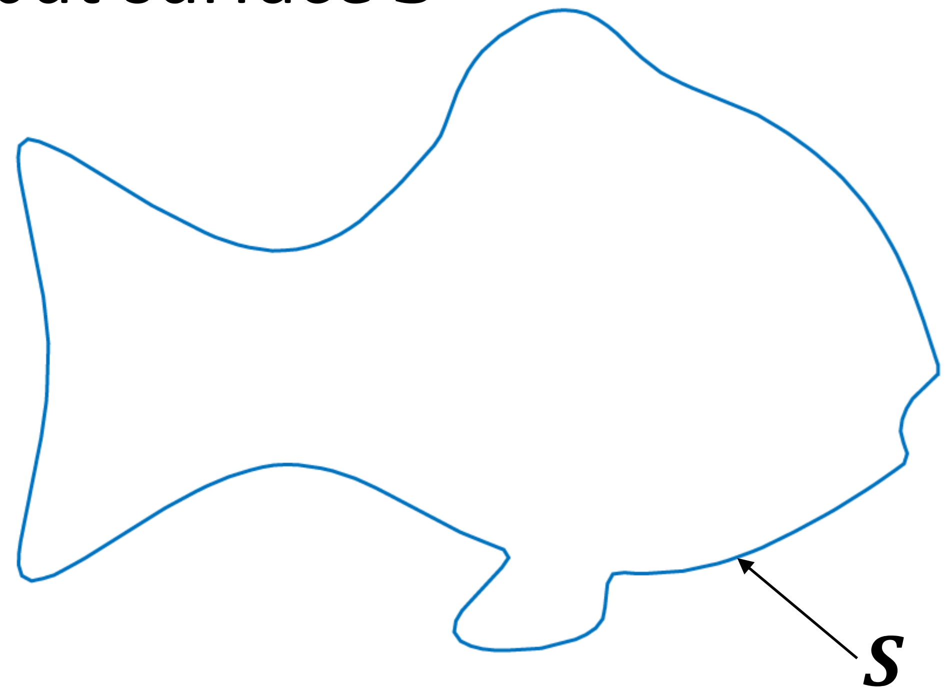
Input and Output

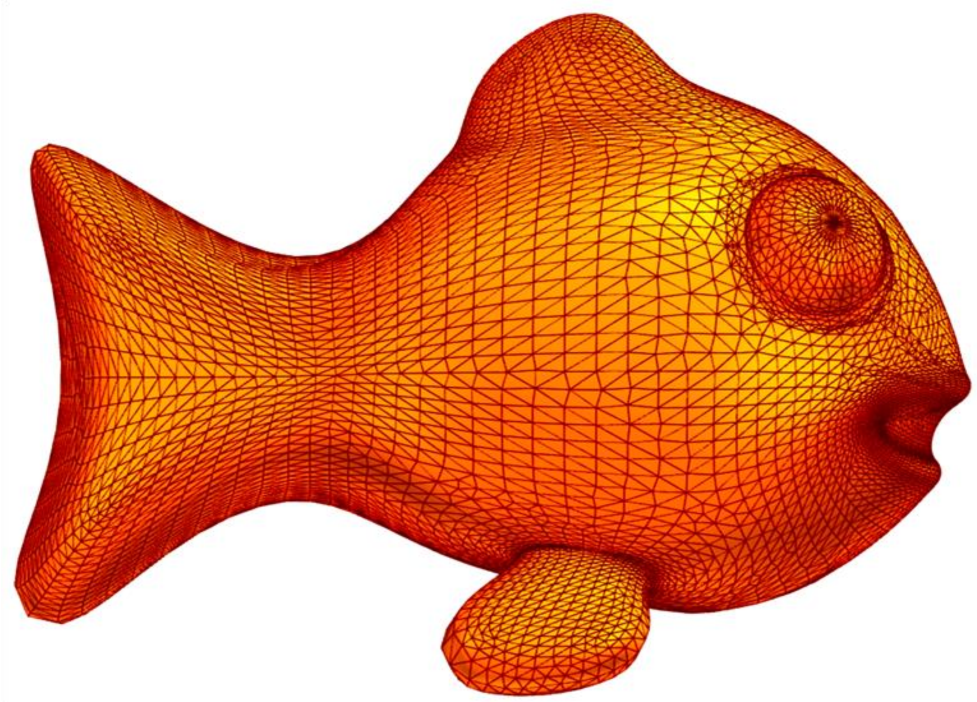
- Input surface S



Input and Output

- input surface $\boldsymbol{S}$

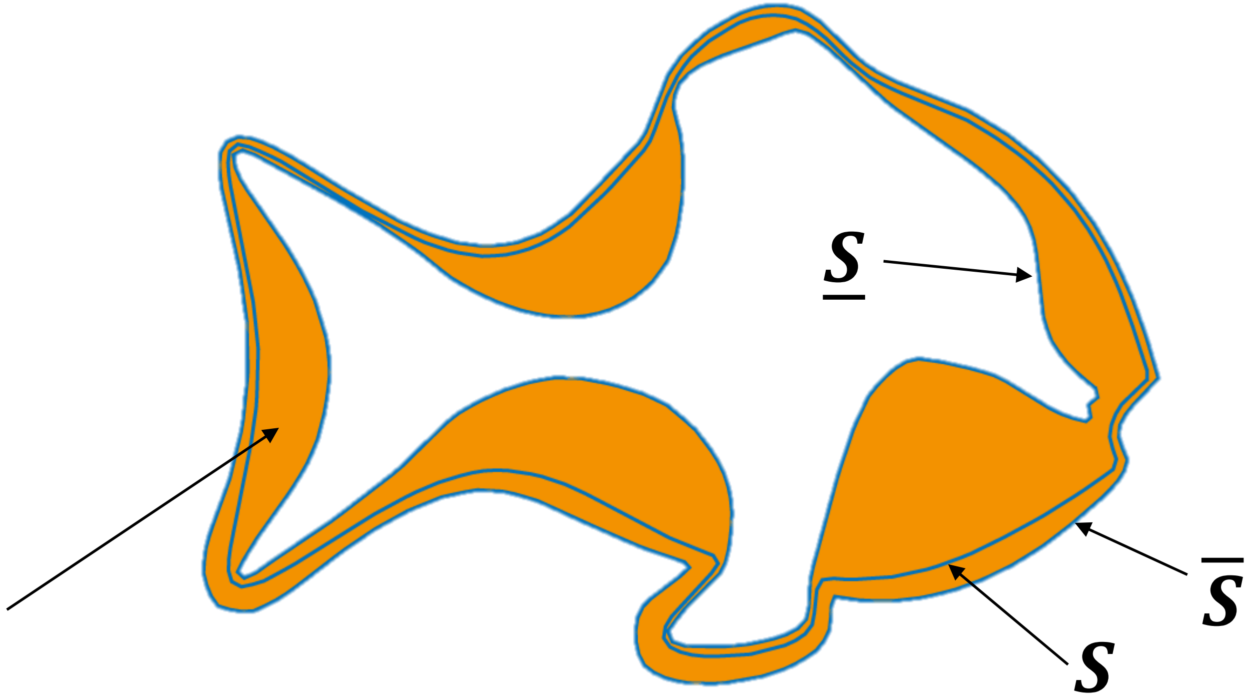




Input and Output

- input surface $\boldsymbol{S}$
- output: two surfaces
  - outer offset surface $\overline{\boldsymbol{S}}$
  - inner offset surface $\underline{\boldsymbol{S}}$

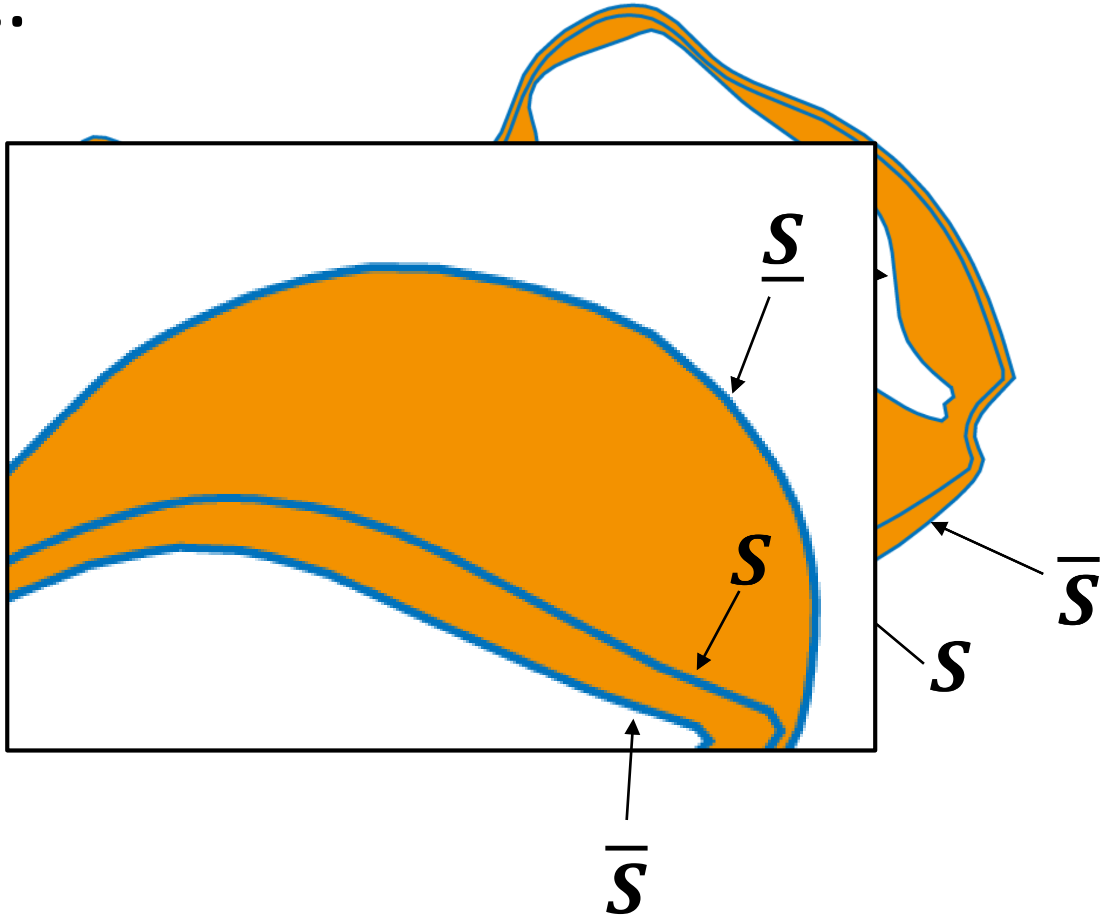
- **solid body** between $\overline{\boldsymbol{S}}$ and $\underline{\boldsymbol{S}}$



Offset Surfaces

- surface deformation by offset:

$$\underline{\boldsymbol{x}}_{\boldsymbol{i}} = \boldsymbol{x}_{\boldsymbol{i}} + \underline{\delta}_i \boldsymbol{v}_{\boldsymbol{i}}$$

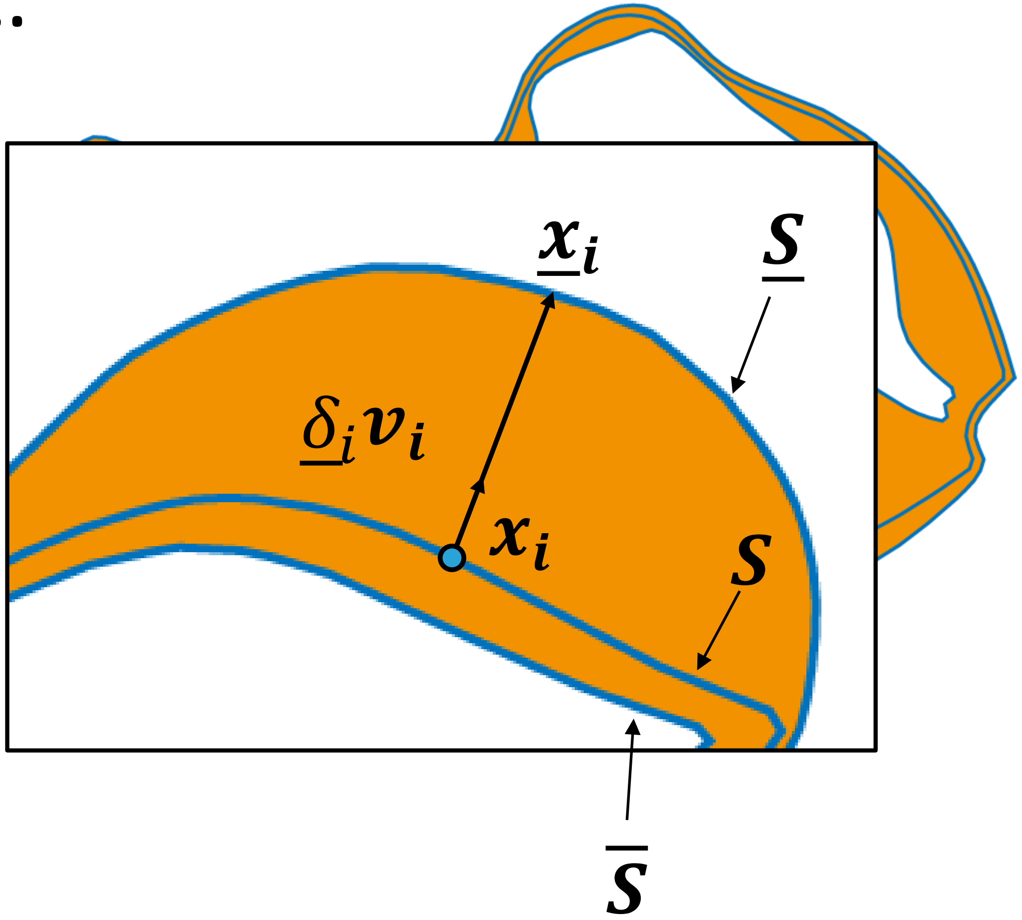


Offset Surfaces

- surface deformation by offset:

$$\underline{\boldsymbol{x}}_i = \boldsymbol{x}_i + \underline{\delta}_i \boldsymbol{v}_i$$

  - for each vertex $\boldsymbol{x}_i$
  - along $\boldsymbol{v}_i$
  - add an individual offset $\underline{\delta}_i$

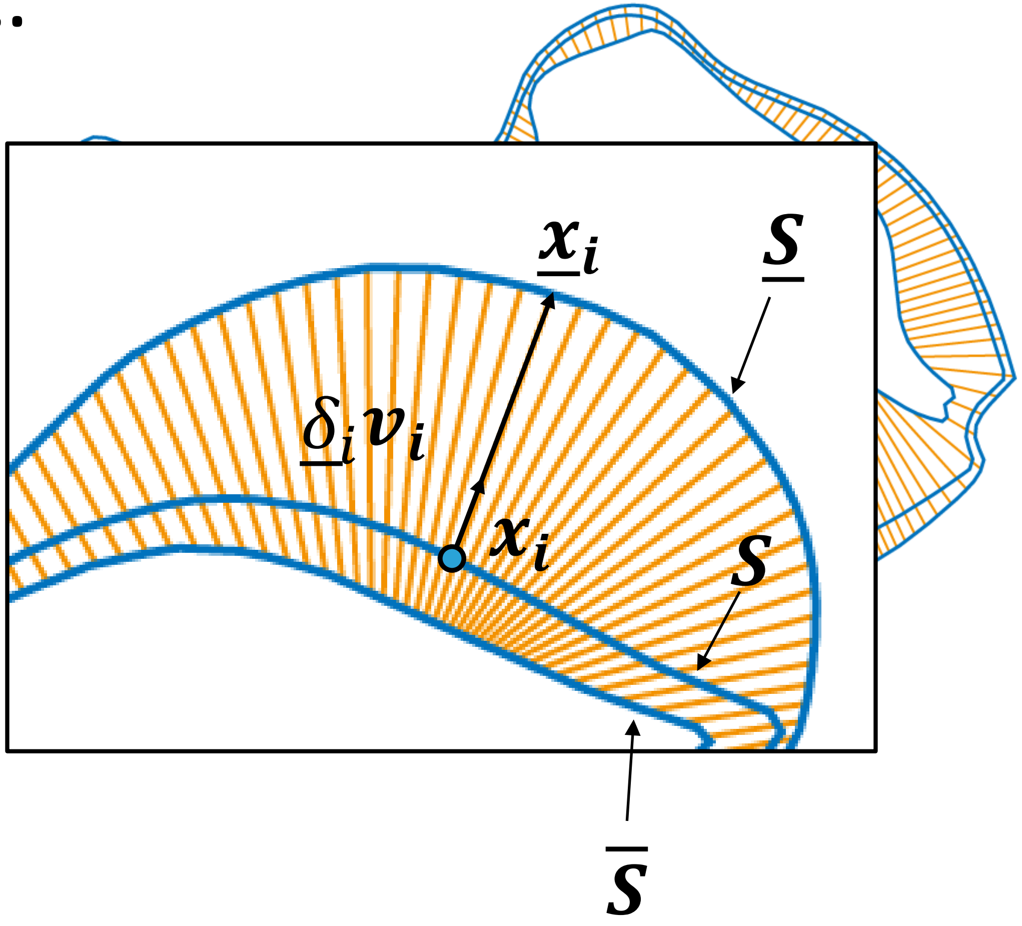


Offset Surfaces

- surface deformation by offset:

$$\underline{\boldsymbol{x}}_i = \boldsymbol{x}_i + \underline{\delta}_i \boldsymbol{v}_i$$

  - for each vertex $\boldsymbol{x}_i$
  - along $\boldsymbol{v}_i$
  - add an individual offset $\underline{\delta}_i$

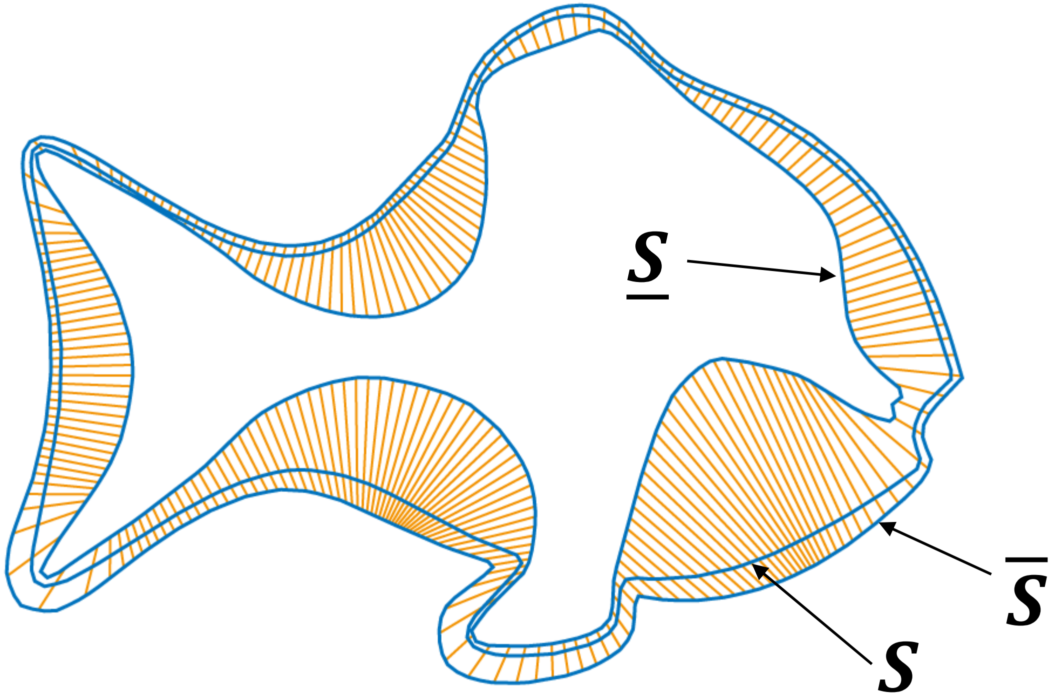


Offset Surfaces

- surface deformation by offset:

$$\underline{\boldsymbol{x}}_i = \boldsymbol{x}_i + \underline{\delta}_i \boldsymbol{v}_i$$

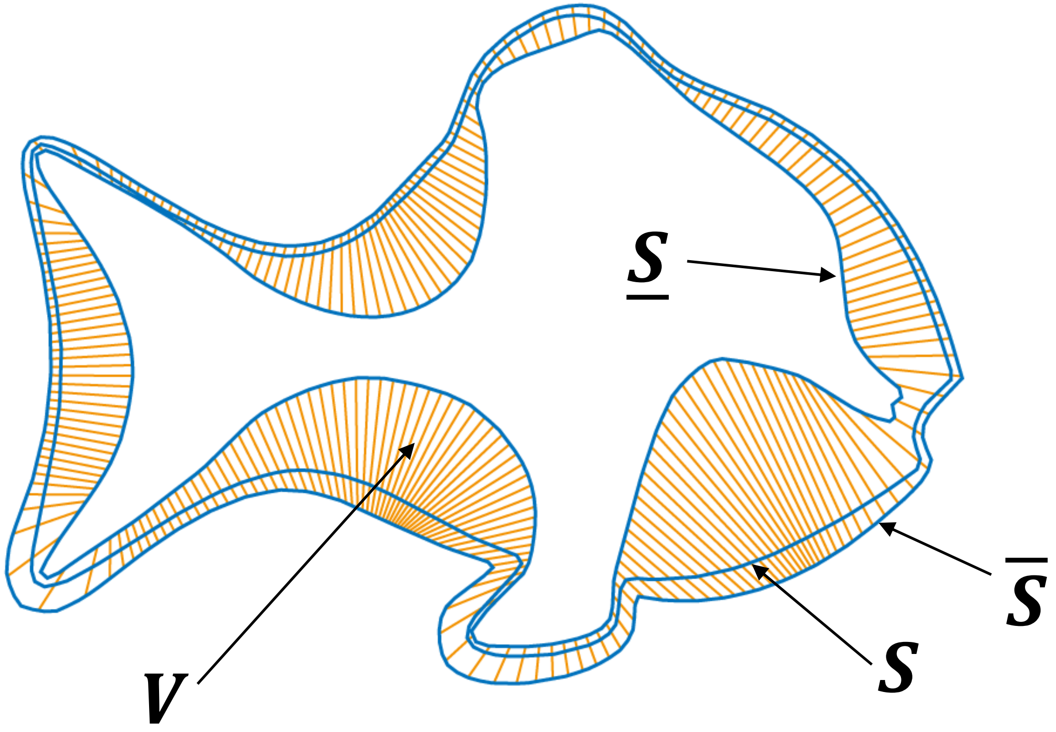
  - for each vertex $\boldsymbol{x}_i$
  - along $\boldsymbol{v}_i$
  - add an individual offset $\underline{\delta}_i$



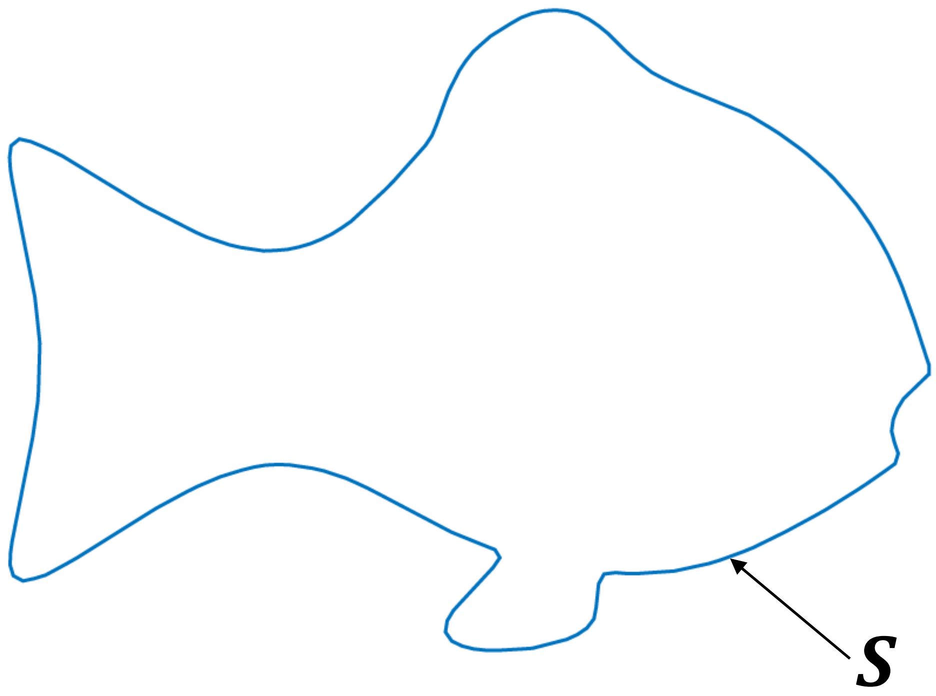
Offset Surfaces

- How far can we offset?

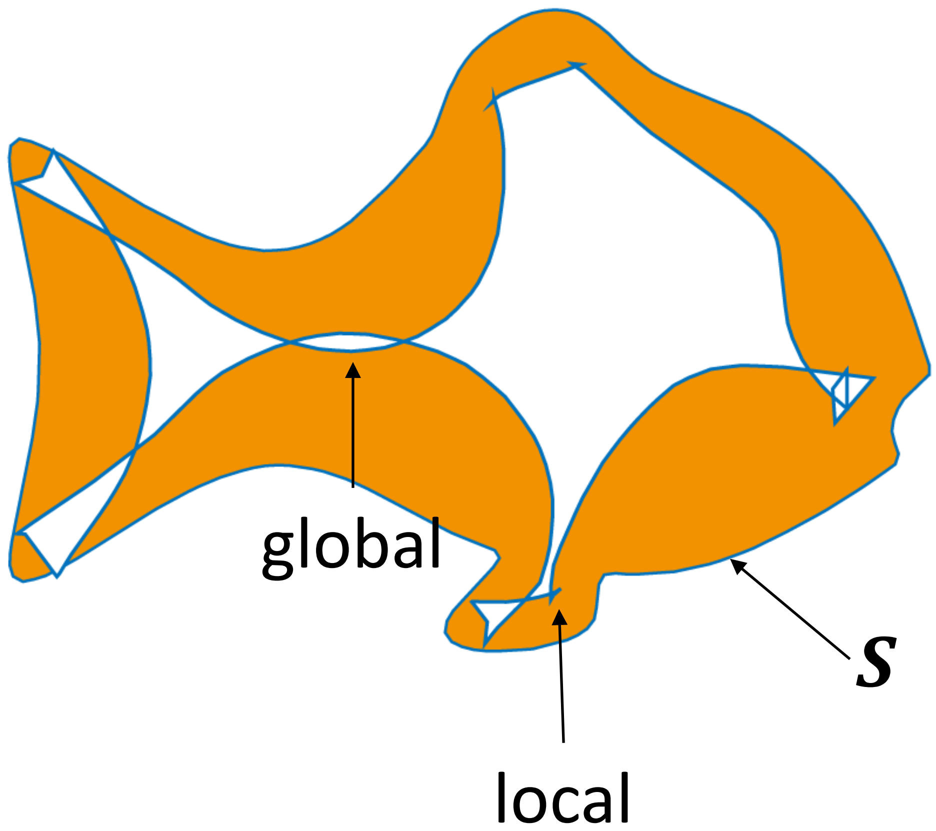
- Along which directions $\boldsymbol{V}$?



Offset Bounds

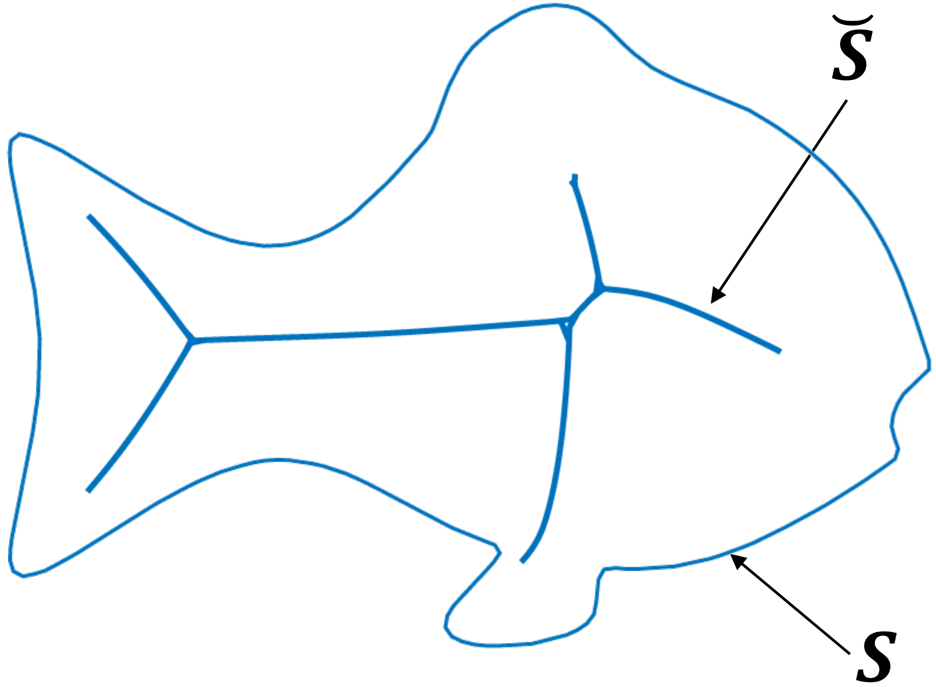


Offset Bounds



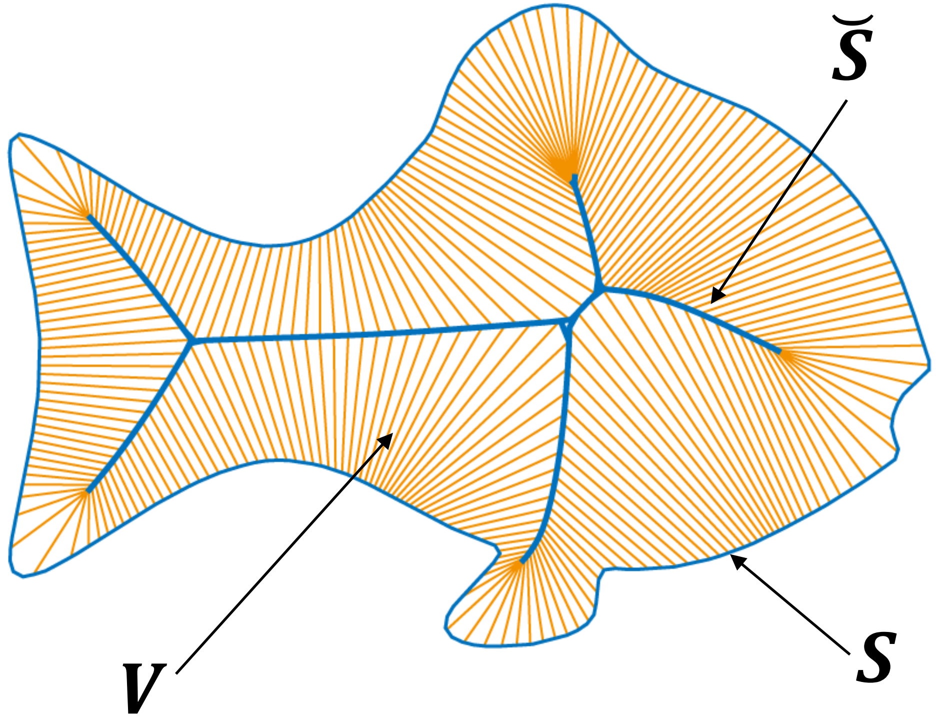
Offset Bounds

- inside: skeleton $\breve{S}$
  - Mean Curvature Flow [Tagliasacchi et al. 2012]



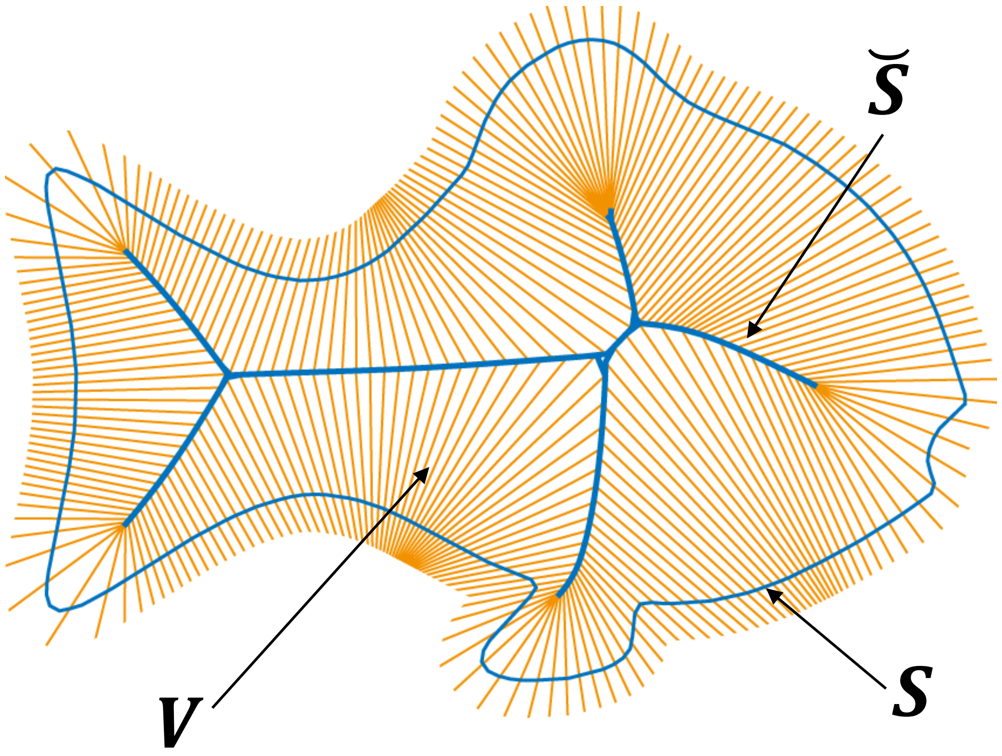
Offset Vectors

- inside: skeleton $\breve{\boldsymbol{S}}$
  - Mean Curvature Flow [Tagliasacchi et al. 2012]
- offset along vectors $\boldsymbol{v_i} \in \boldsymbol{V}$



Offset Vectors and Bounds

- inside: skeleton $\breve{\boldsymbol{S}}$
  - Mean Curvature Flow [Tagliasacchi et al. 2012]
- offset along vectors $\boldsymbol{v_i} \in \boldsymbol{V}$
- outside a constant max. value



Shape Optimization Problem

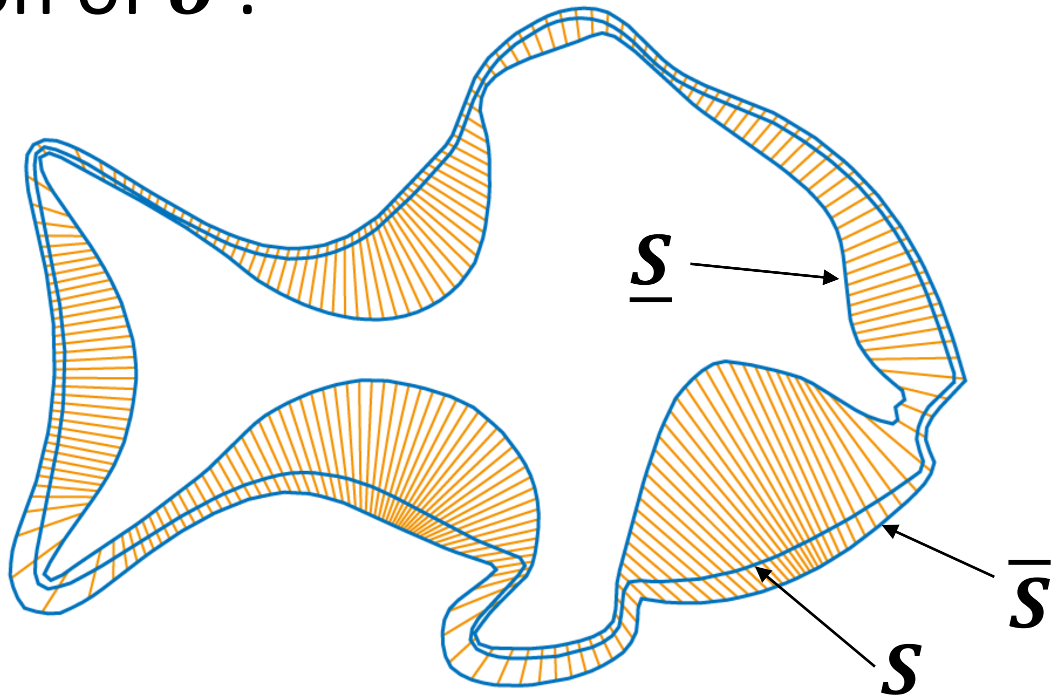
- minimize objective f as a function of $\boldsymbol{\delta}$:

$$\min_{\boldsymbol{\delta}} f\,(\boldsymbol{\delta}) \;\; \text{s.t.} \;\; g(\boldsymbol{\delta})$$

- subject to constraints $g(\boldsymbol{\delta})$

- for example:
  - f := make shape float

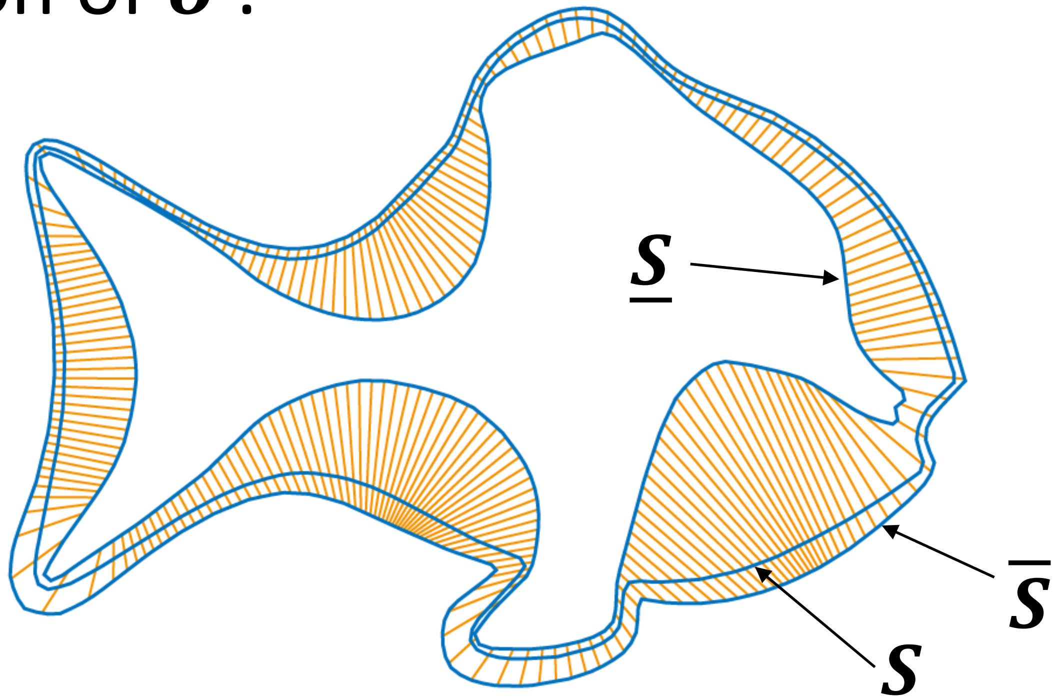
    subject to
  - g := keep upright orientation



Shape Optimization Problem

- minimize objective f as a function of $\boldsymbol{\delta}$:

$$\min_{\boldsymbol{\delta}} f\,(\boldsymbol{\delta}) \quad \rightarrow n \text{ unknowns}$$



- issues:
  - problem is huge for large meshes

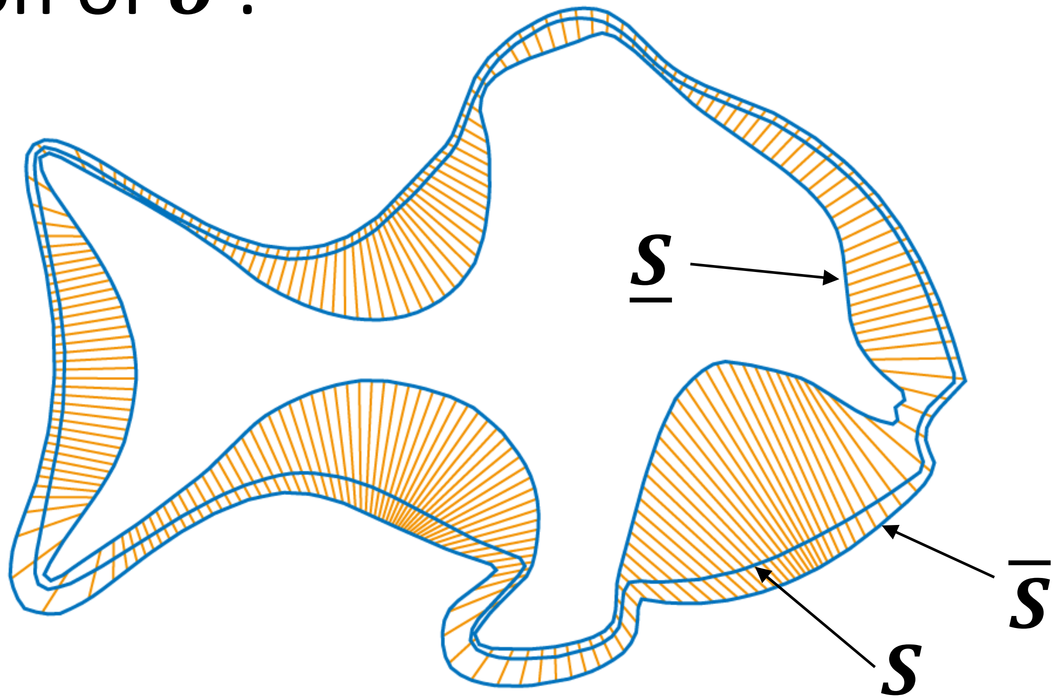
  → **scales very badly**
  - problem is underdetermined

  → **there exist many solutions** (regularization needed)

Shape Optimization Problem

- minimize objective f as a function of $\boldsymbol{\delta}$:

ergo: formulation is not suitable for practice



- issues:
  - problem is huge for large meshes

  → **scales very badly**
  - problem is underdetermined

  → **there exist many solutions** (regularization needed)

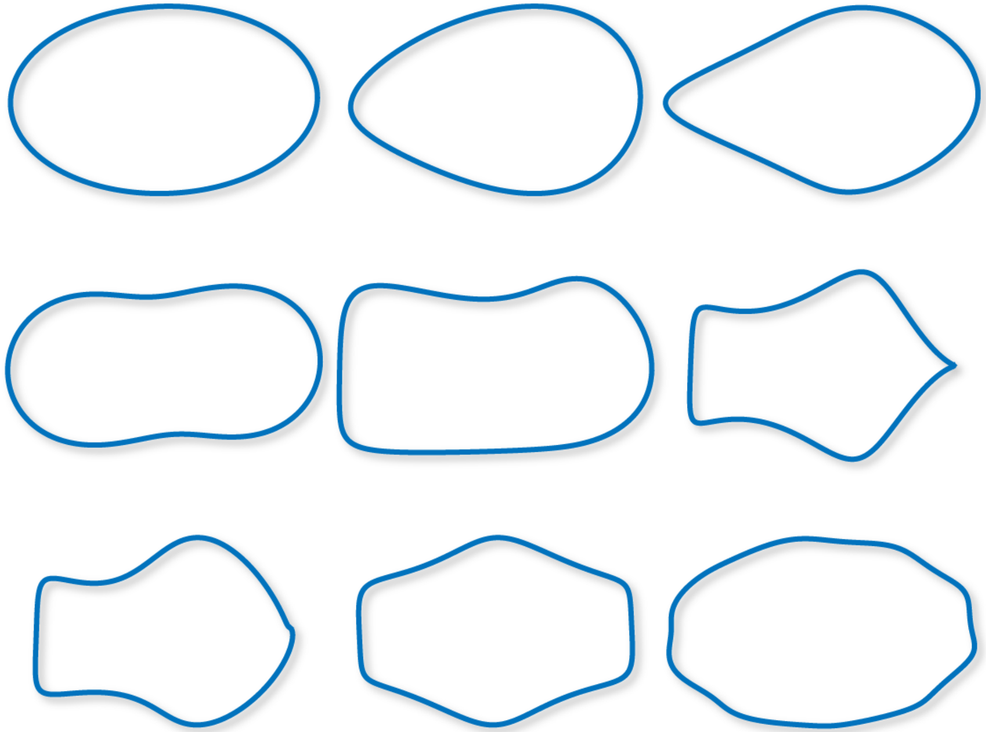
Order Reduction

Order Reduction

- order reduction:
  - lower the dimensionality while preserving input-output behavior

- idea:
  - project problem onto a lower dimensional space

- → Manifold Harmonics



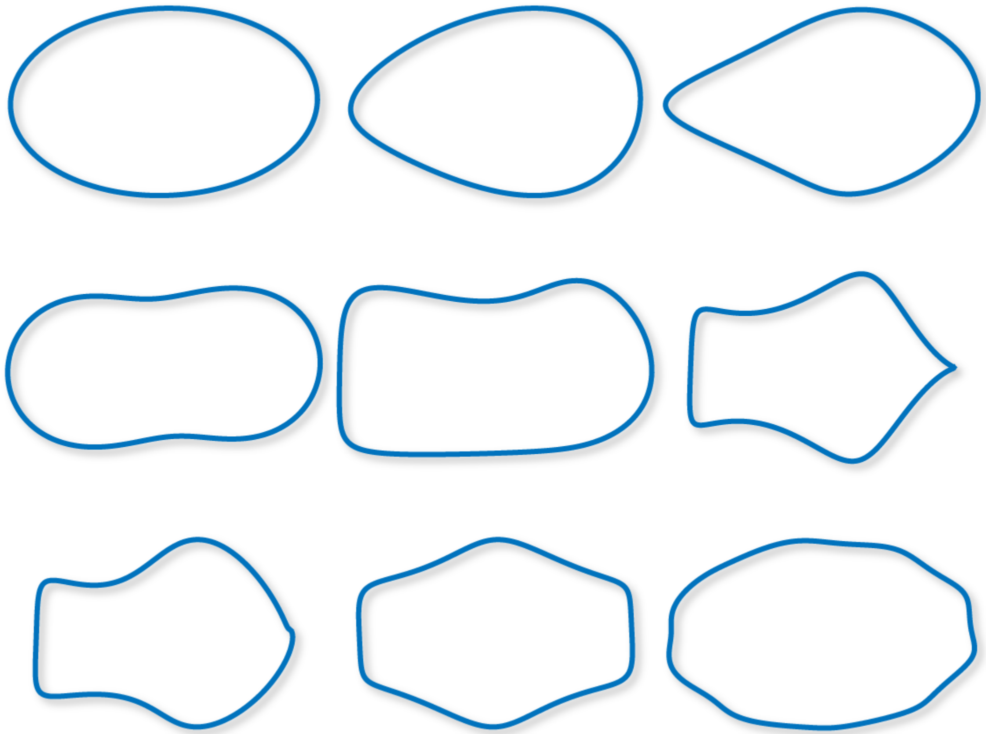
Manifold Harmonics

- generalization of the **Fourier Transform** for scalar functions on surfaces

- diagonalization of the Laplacian matrix

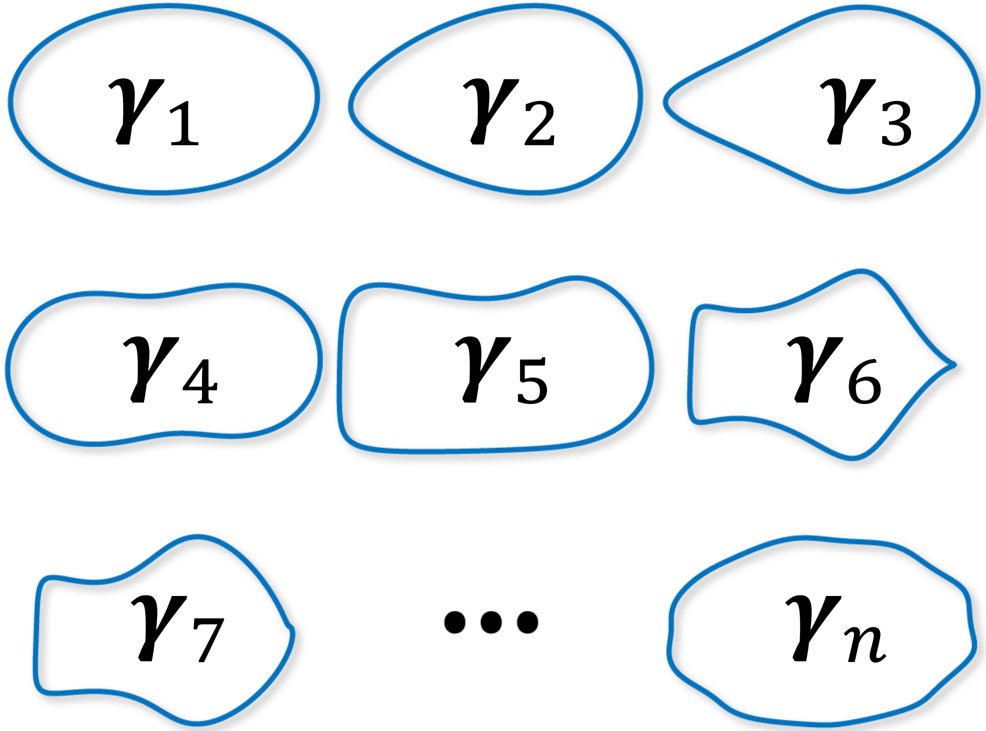
  → Spectral Theorem:

$$\mathbf{L} = \mathbf{\Gamma \Lambda \Gamma^T}$$



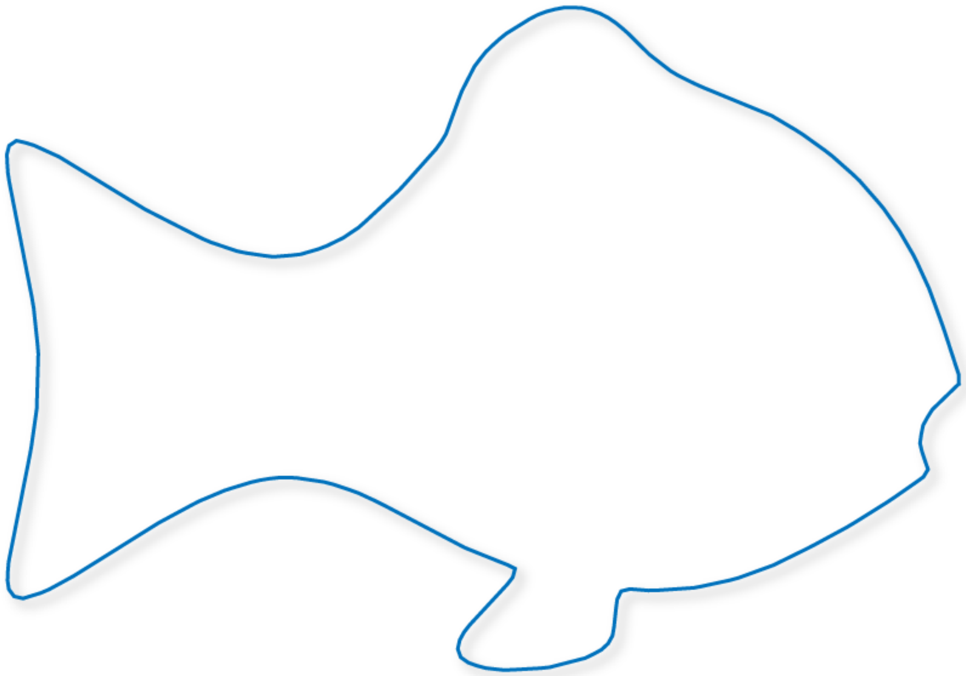
Manifold Harmonics

- eigenfunctions
  - $\mathbf{\Gamma} = [\,\boldsymbol{\gamma}_1\, \boldsymbol{\gamma}_2\ \dots\ \boldsymbol{\gamma}_n]$
- shape can be transformed to
  - $\widetilde{\boldsymbol{X}} = \mathbf{\Gamma}^{\mathrm{T}} \boldsymbol{X}$



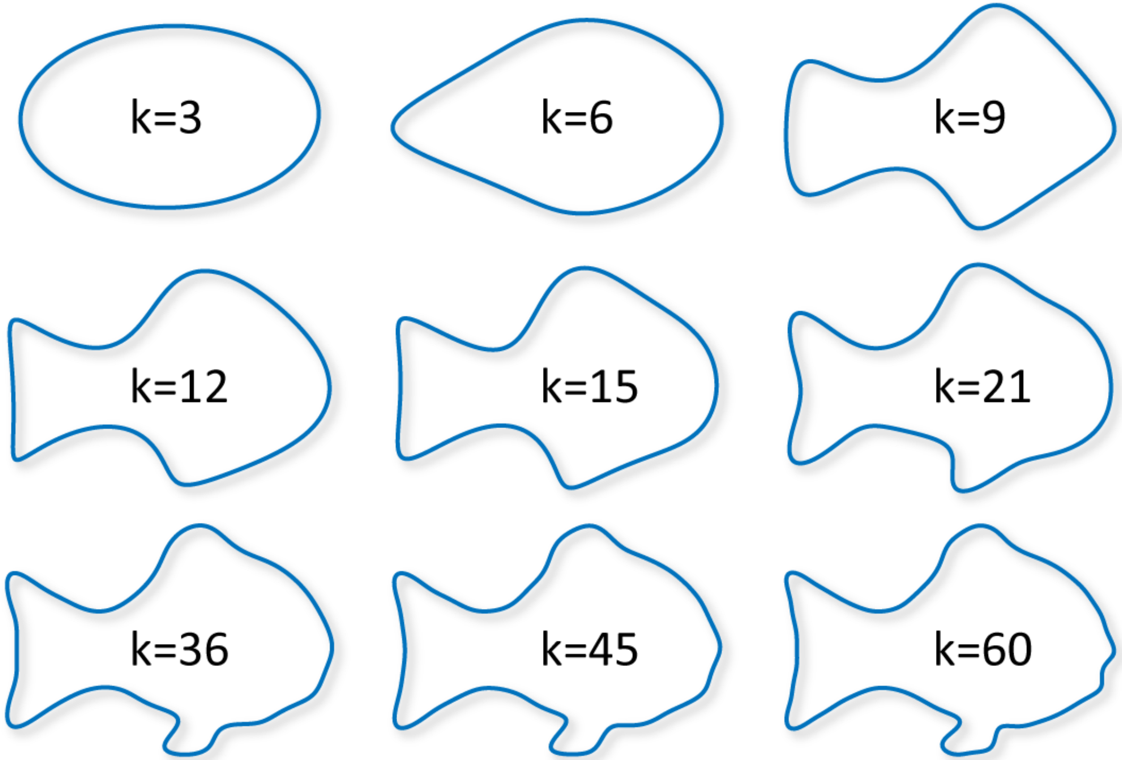
Manifold Harmonics

- eigenfunctions
  - $\boldsymbol{\Gamma} = [\,\boldsymbol{\gamma}_1\,\boldsymbol{\gamma}_2\ \ldots\ \boldsymbol{\gamma}_n]$
- shape can be transformed to
  - $\widetilde{\boldsymbol{X}} = \boldsymbol{\Gamma}^{\mathrm{T}}\boldsymbol{X}$
- reconstruction
  - $\boldsymbol{X}_k = \boldsymbol{\Gamma}_k\ \widetilde{\boldsymbol{X}}_k$
  - with $\boldsymbol{\Gamma}_k = [\,\boldsymbol{\gamma}_1\,\boldsymbol{\gamma}_2\ \ldots\ \boldsymbol{\gamma}_k]$



Manifold Harmonics

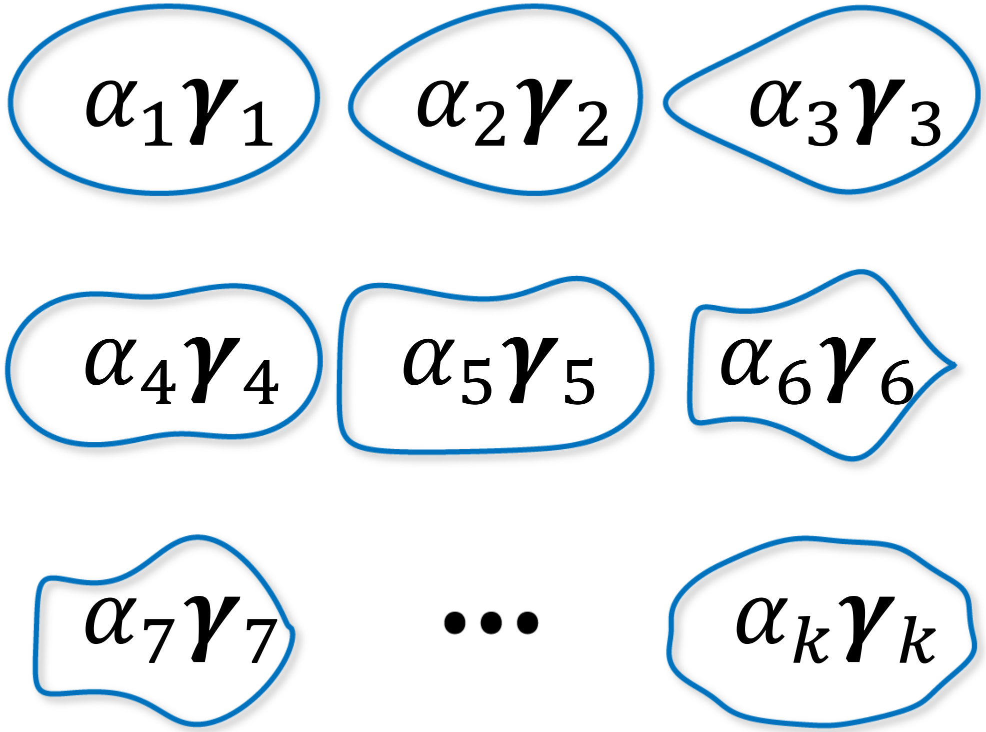
- eigenfunctions
  - $\boldsymbol{\Gamma} = [\,\boldsymbol{\gamma}_1\,\boldsymbol{\gamma}_2\;\dots\;\boldsymbol{\gamma}_n]$
- shape can be transformed to
  - $\widetilde{\boldsymbol{X}} = \boldsymbol{\Gamma}^{\mathrm{T}}\boldsymbol{X}$
- reconstruction
  - $\boldsymbol{X}_k = \boldsymbol{\Gamma}_k\;\widetilde{\boldsymbol{X}}_k$
  - with $\boldsymbol{\Gamma}_k = [\,\boldsymbol{\gamma}_1\,\boldsymbol{\gamma}_2\;\dots\;\boldsymbol{\gamma}_k]$



Order Reduction

- **project unknown offsets** $\boldsymbol{\delta} = [\delta_1, \delta_2, \ldots, \delta_n]^T$ onto $\boldsymbol{\Gamma}_k$:

$$\boldsymbol{\delta} = \boldsymbol{\Gamma}_k \boldsymbol{\alpha} = \sum$$



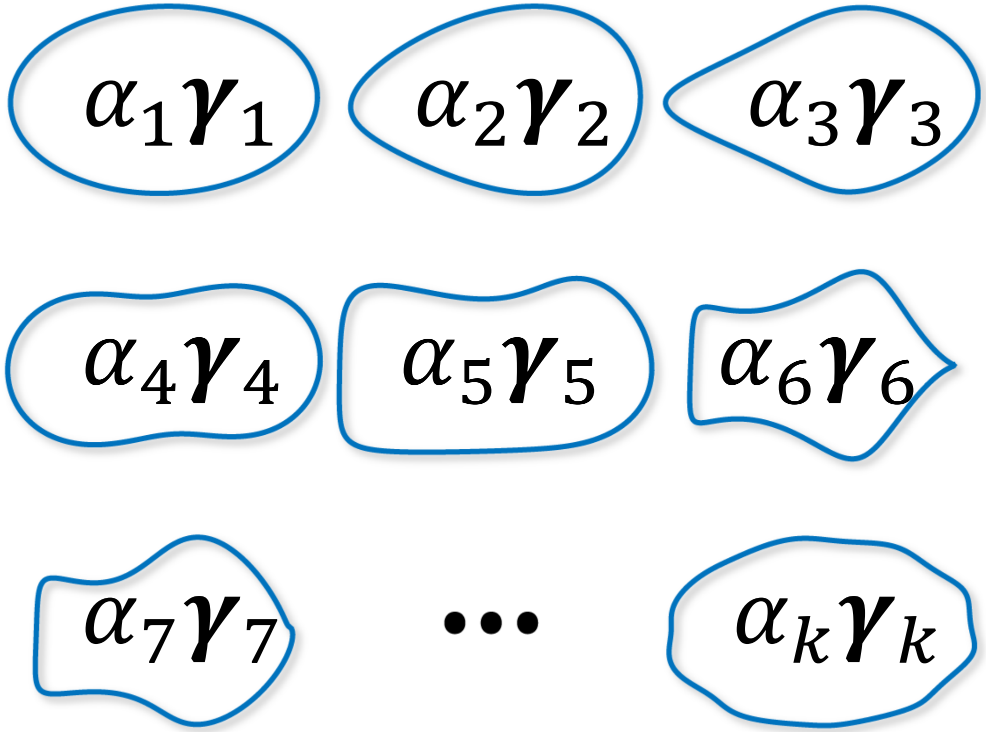
- vector $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \ldots, \alpha_k]^T$ now contains the unknowns!

Order Reduction

- **project unknown offsets** $\boldsymbol{\delta} = [\delta_1, \delta_2, \ldots, \delta_n]^T$ onto $\boldsymbol{\Gamma}_k$:

$$\bar{\boldsymbol{x}}_i = \boldsymbol{x}_i + \delta_i \boldsymbol{v}_i$$

$$\bar{\boldsymbol{x}}_i = \boldsymbol{x}_i + \sum_{j=1}^{k} \alpha_j \gamma_{ij} \boldsymbol{v}_i$$

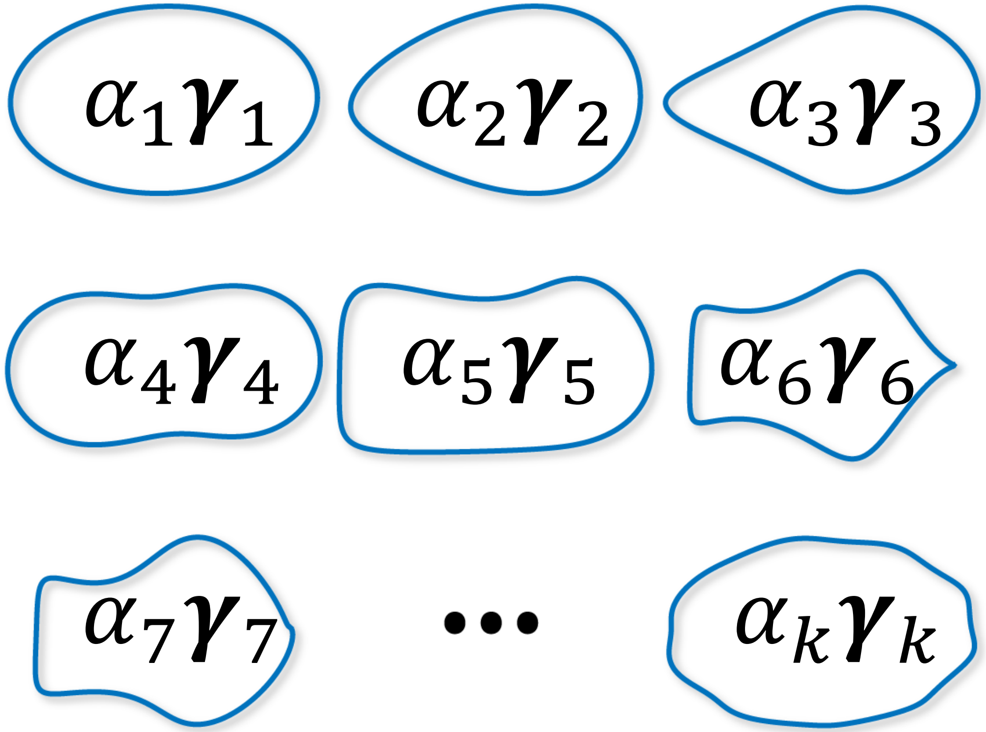


Reduced Shape Optimization Problem

- minimize objective f as a function of $\boldsymbol{\alpha}$:

$$\min_{\boldsymbol{\alpha}} f\,(\boldsymbol{\alpha})$$

- (subject to constraints)

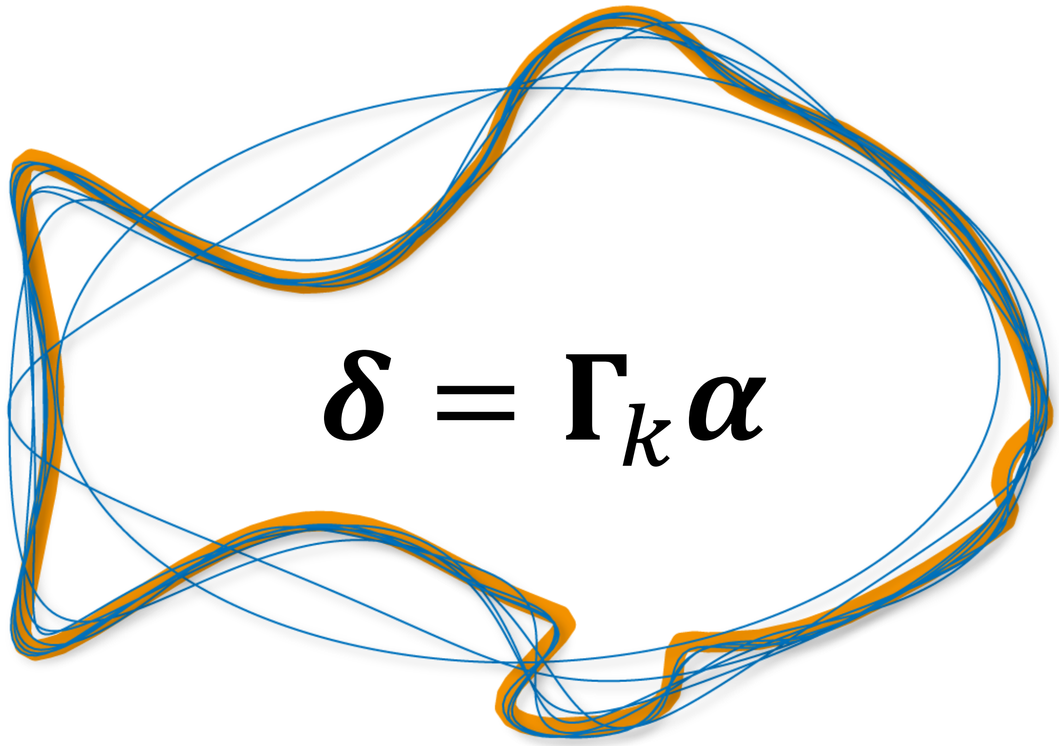


Reduced Shape Optimization Problem

- minimize objective f as a function of $\boldsymbol{\alpha}$:

$$\min_{\boldsymbol{\alpha}} f\,(\boldsymbol{\alpha}) \;\rightarrow k \text{ unknowns, } k \ll n$$

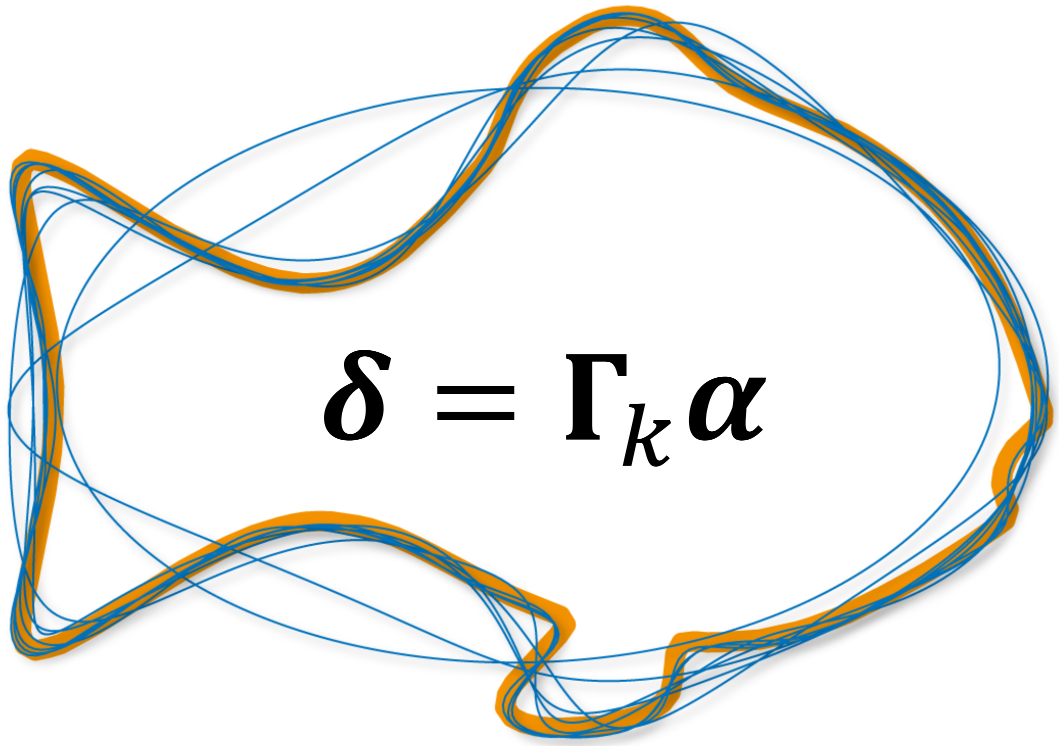
We **deform only** the **low-frequencies** and leave high-frequency details untouched!



Reduced Shape Optimization Problem

- minimize objective f as a function of $\boldsymbol{\alpha}$:

$$\min_{\boldsymbol{\alpha}} f\,(\boldsymbol{\alpha}) \;\rightarrow k \text{ unknowns, } k \ll n$$



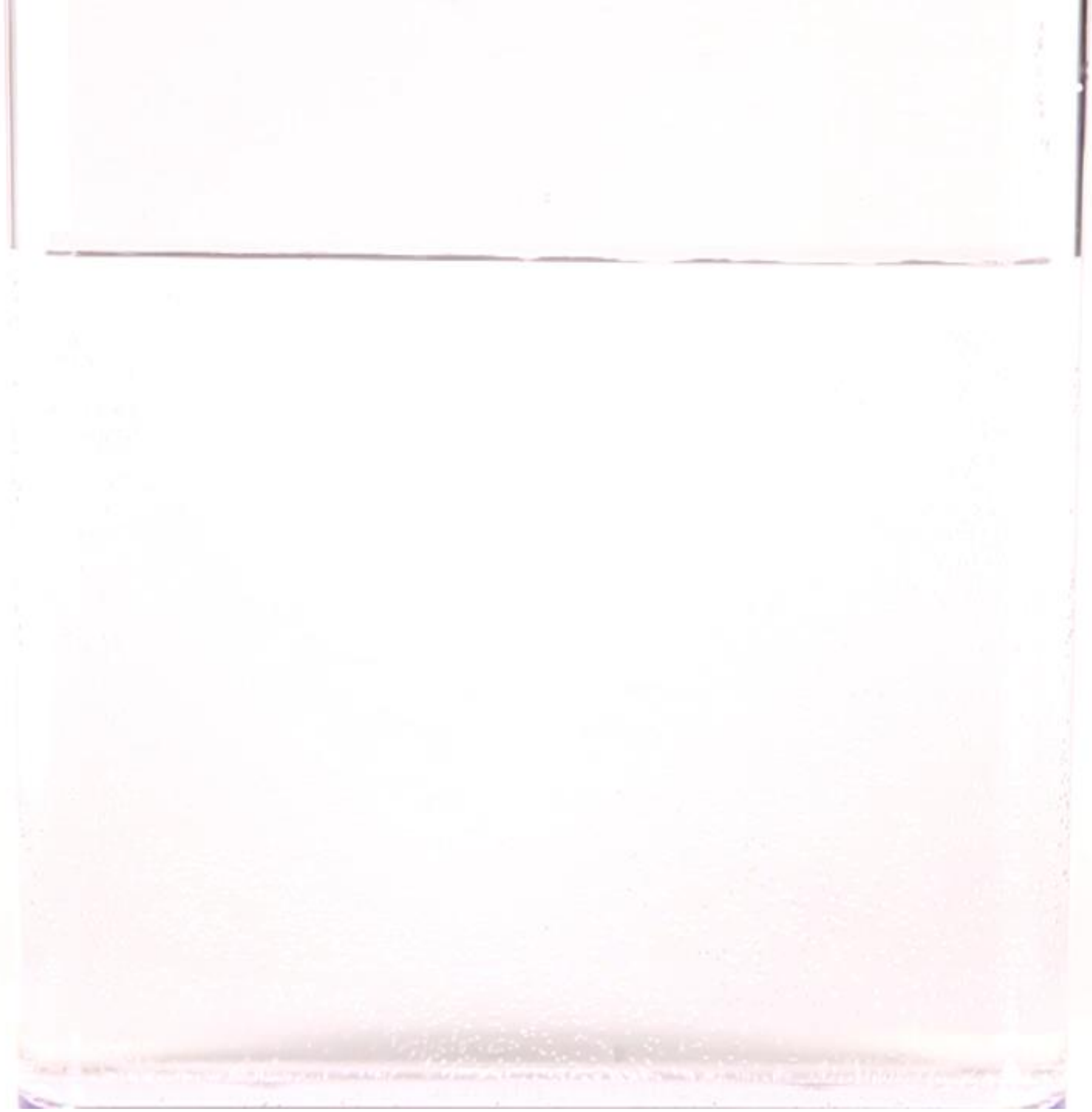
→ independent of mesh resolution

→ implicit regularization

→ numerically stable

→ easy to implement

Applications



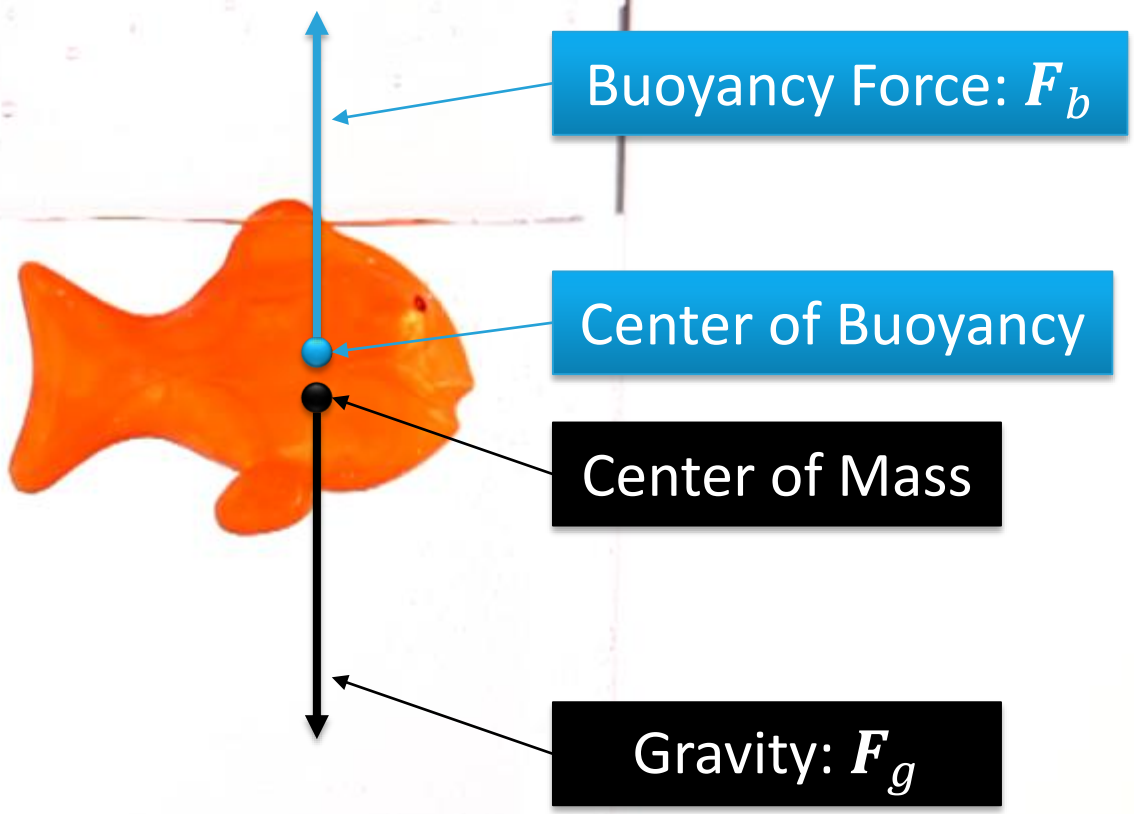
Equilibrium

$$F_g = F_b$$

↓

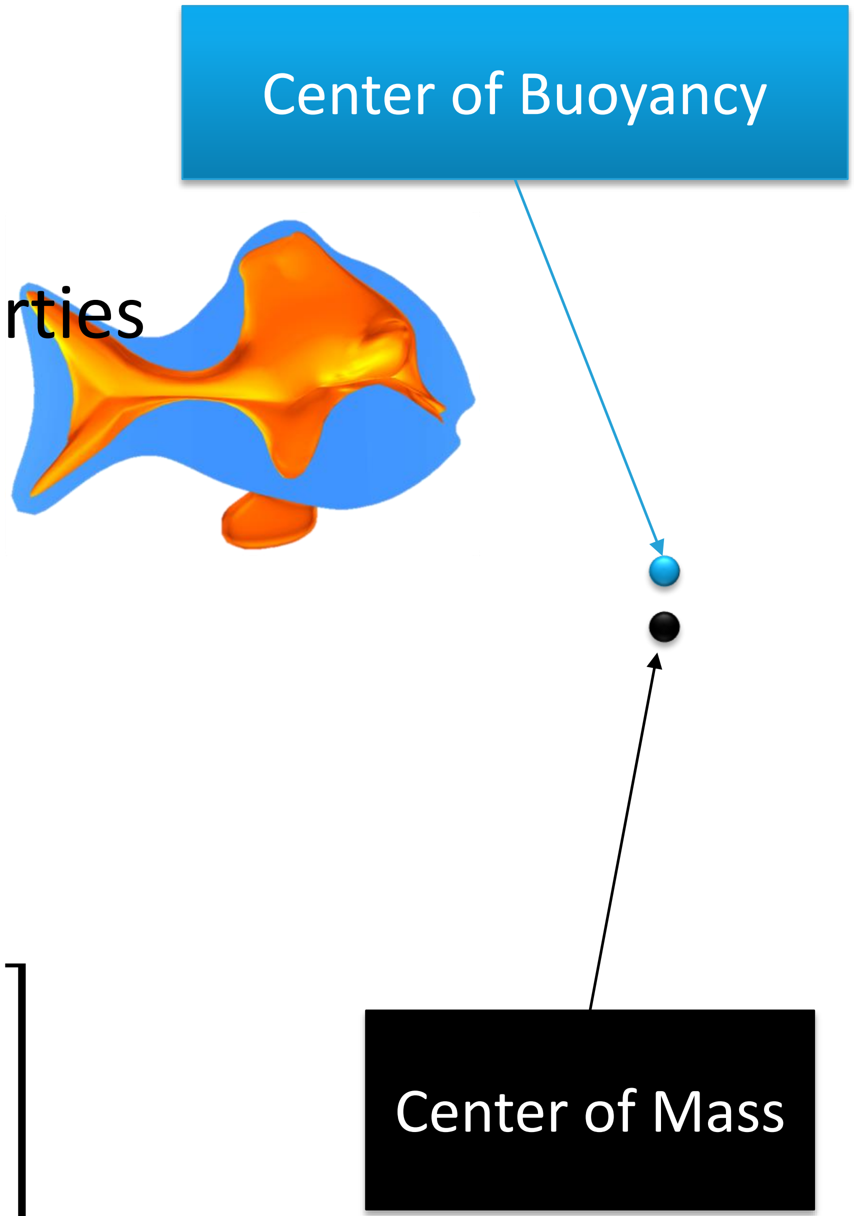
Mass Properties

$$P(S)$$



Applications

- Gauss' Divergence Theorem
  - allows us to compute mass properties as a function of the surface



$$\boldsymbol{P}_{\boldsymbol{m}}(\boldsymbol{S}) = M$$

$$\boldsymbol{P}_{\boldsymbol{x,y,z}}(\boldsymbol{S}) = \boldsymbol{CoM} = [c_x \quad c_y \quad c_z]^{\boldsymbol{T}}$$

$$\boldsymbol{P}_{\boldsymbol{x^2,xy,\ldots,z^2}}(\boldsymbol{S}) = \boldsymbol{I} = \begin{bmatrix} I_{x^2} & I_{xy} & I_{xz} \\ I_{xy} & I_{y^2} & I_{yz} \\ I_{xz} & I_{yz} & I_{z^2} \end{bmatrix}$$

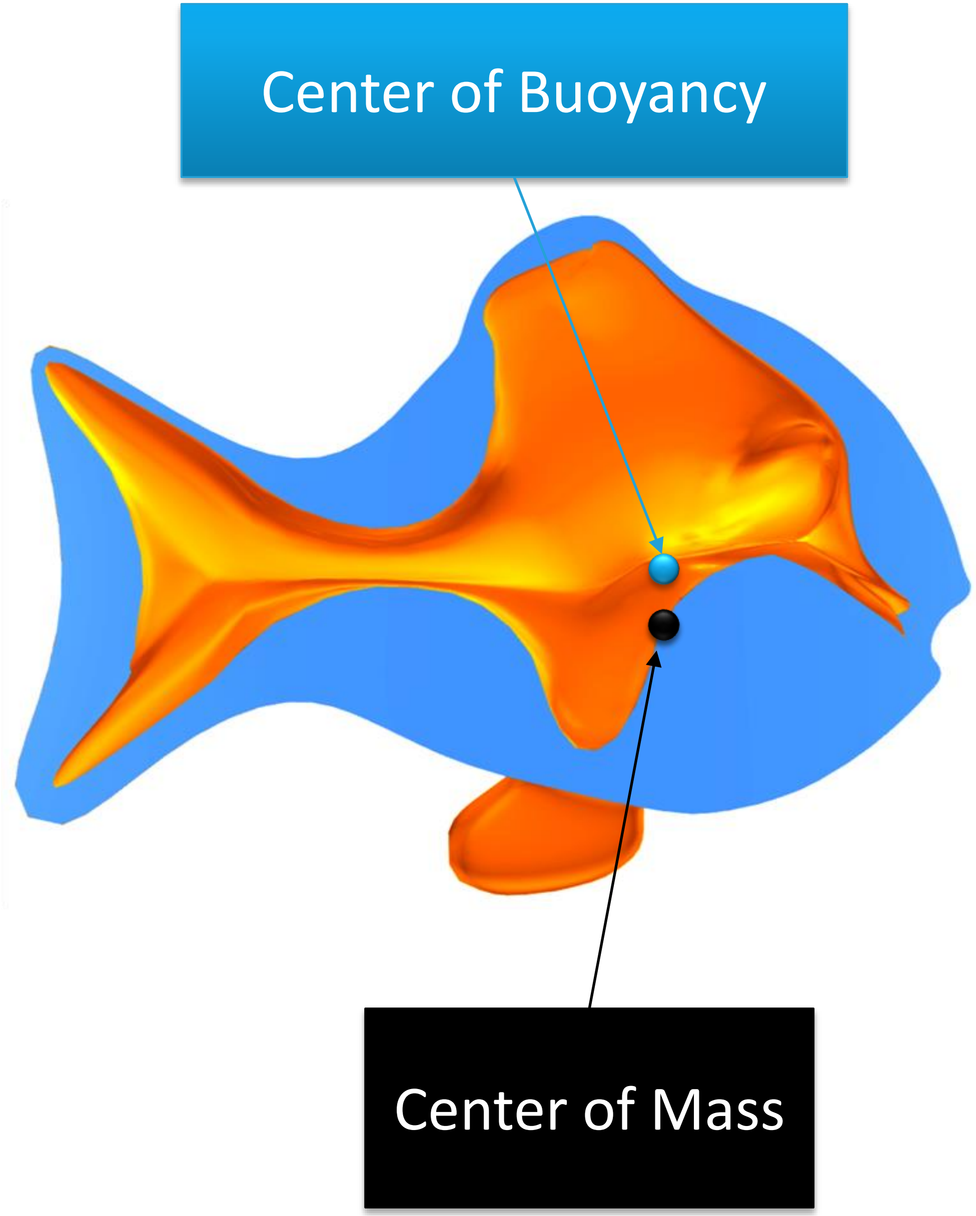
Applications

- optimization problem

$$\min_{\boldsymbol{\alpha}} f\left(\boldsymbol{P}\left(\boldsymbol{S}\left(\boldsymbol{\delta}(\boldsymbol{\alpha})\right)\right)\right)$$

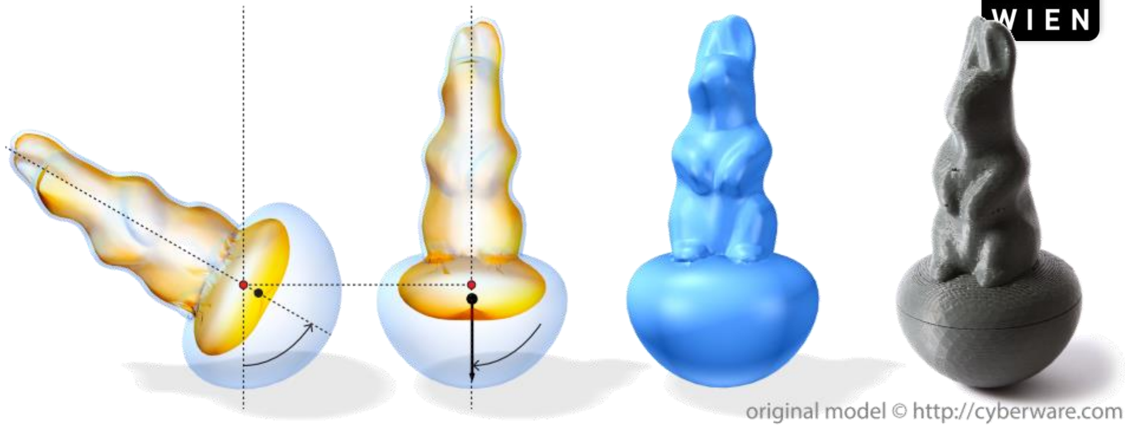
- an analytical gradient

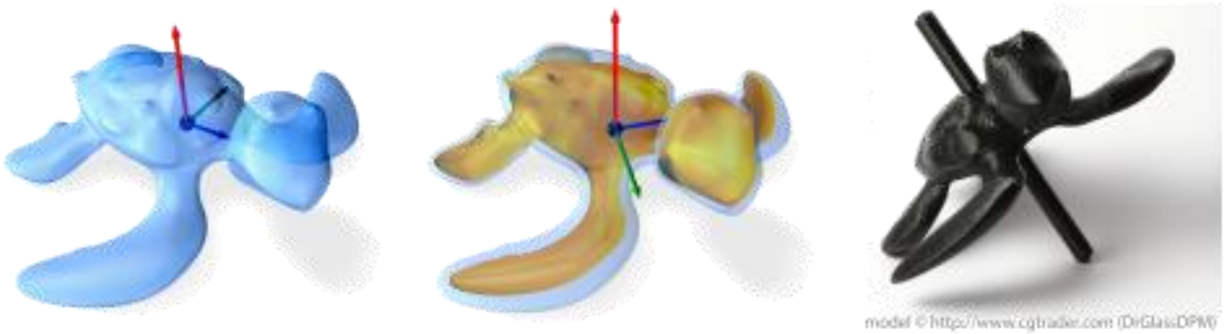
$$\nabla f = \frac{\partial f}{\partial \boldsymbol{P}} \frac{\partial \boldsymbol{P}}{\partial \boldsymbol{S}} \frac{\partial \boldsymbol{S}}{\partial \boldsymbol{\delta}} \frac{\partial \boldsymbol{\delta}}{\partial \boldsymbol{\alpha}}$$



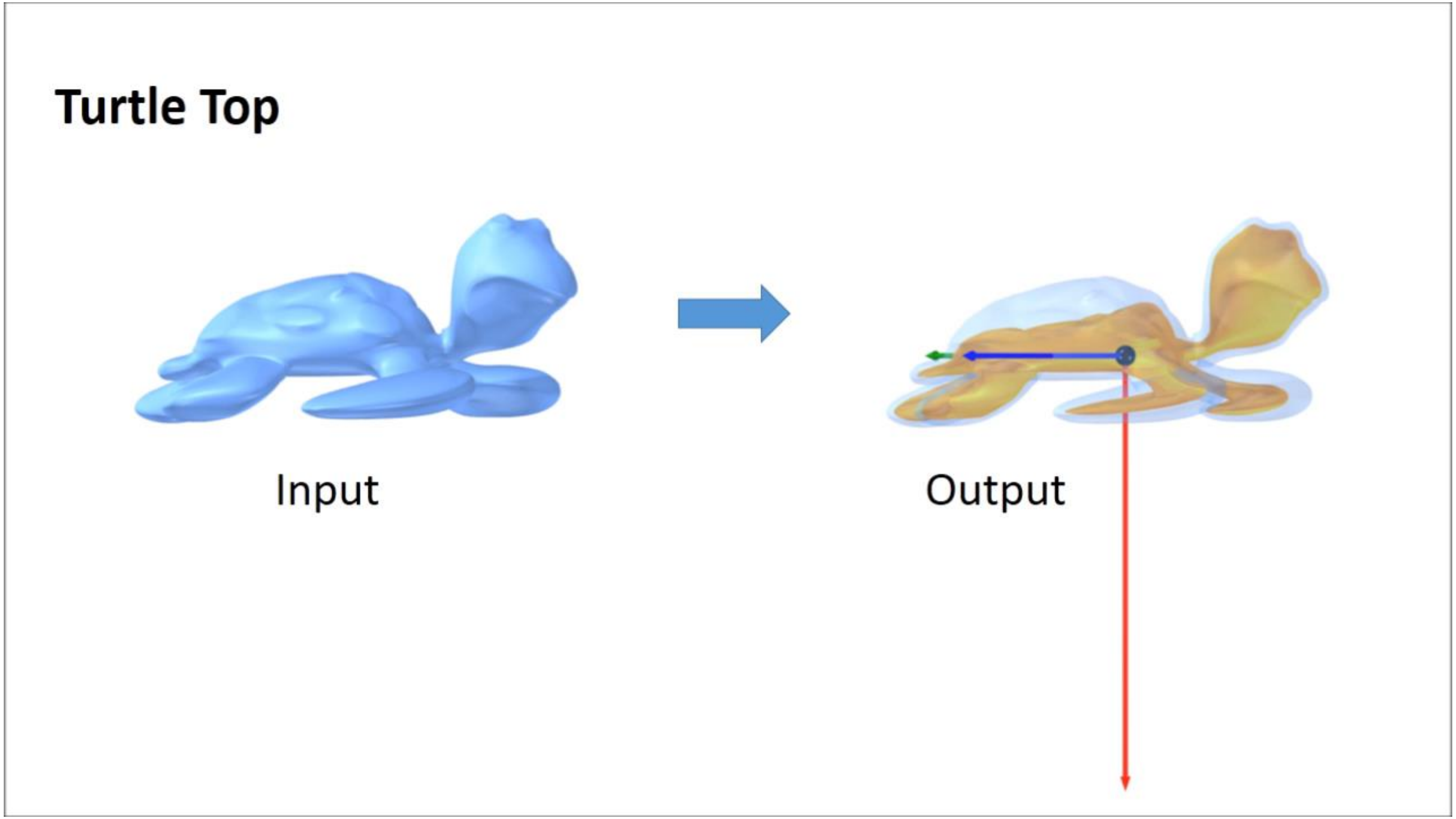
Applications

- static stability
- monostatic stability
- rotational stability
- static stability under storage
- volume and buoyancy

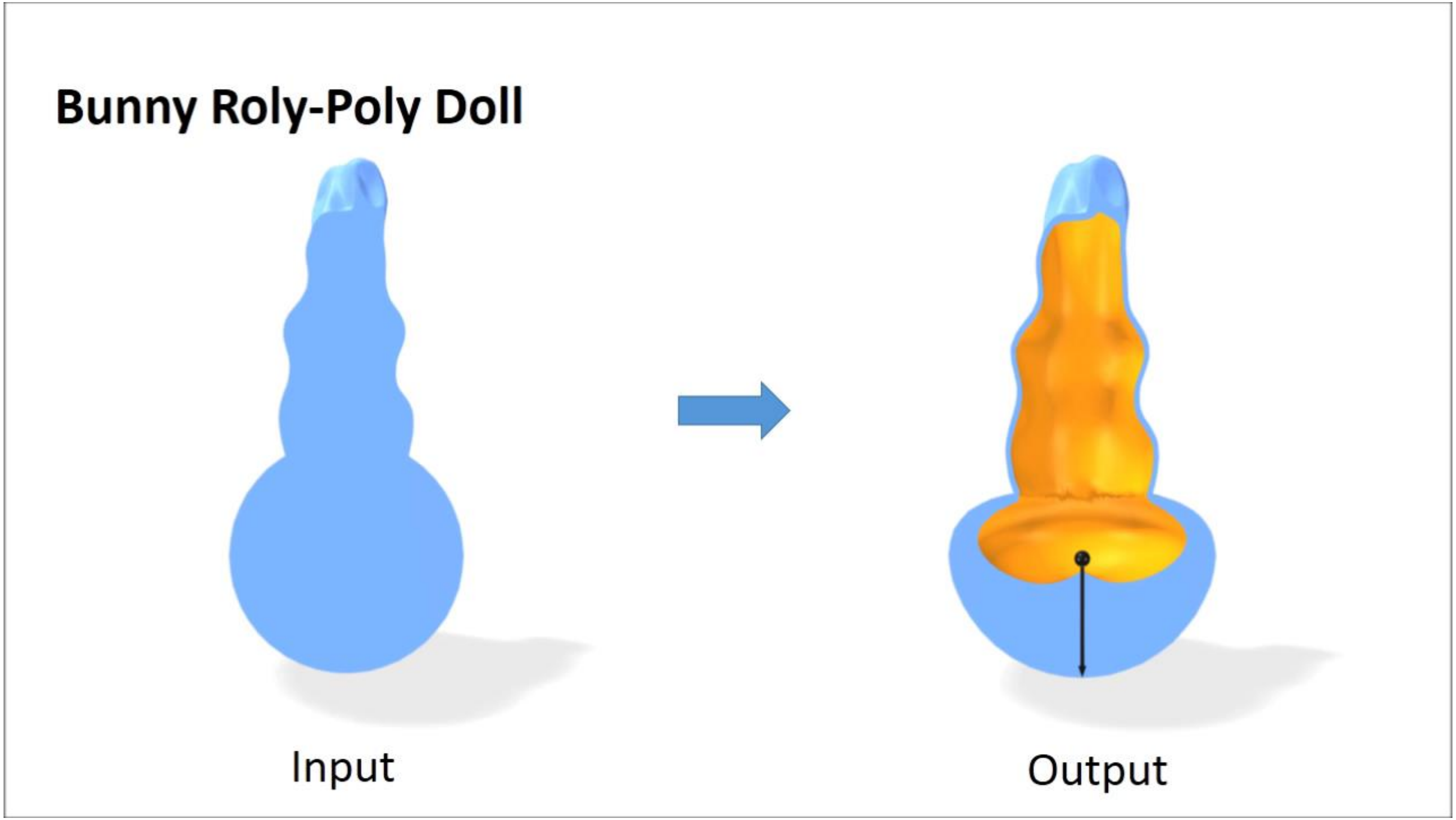




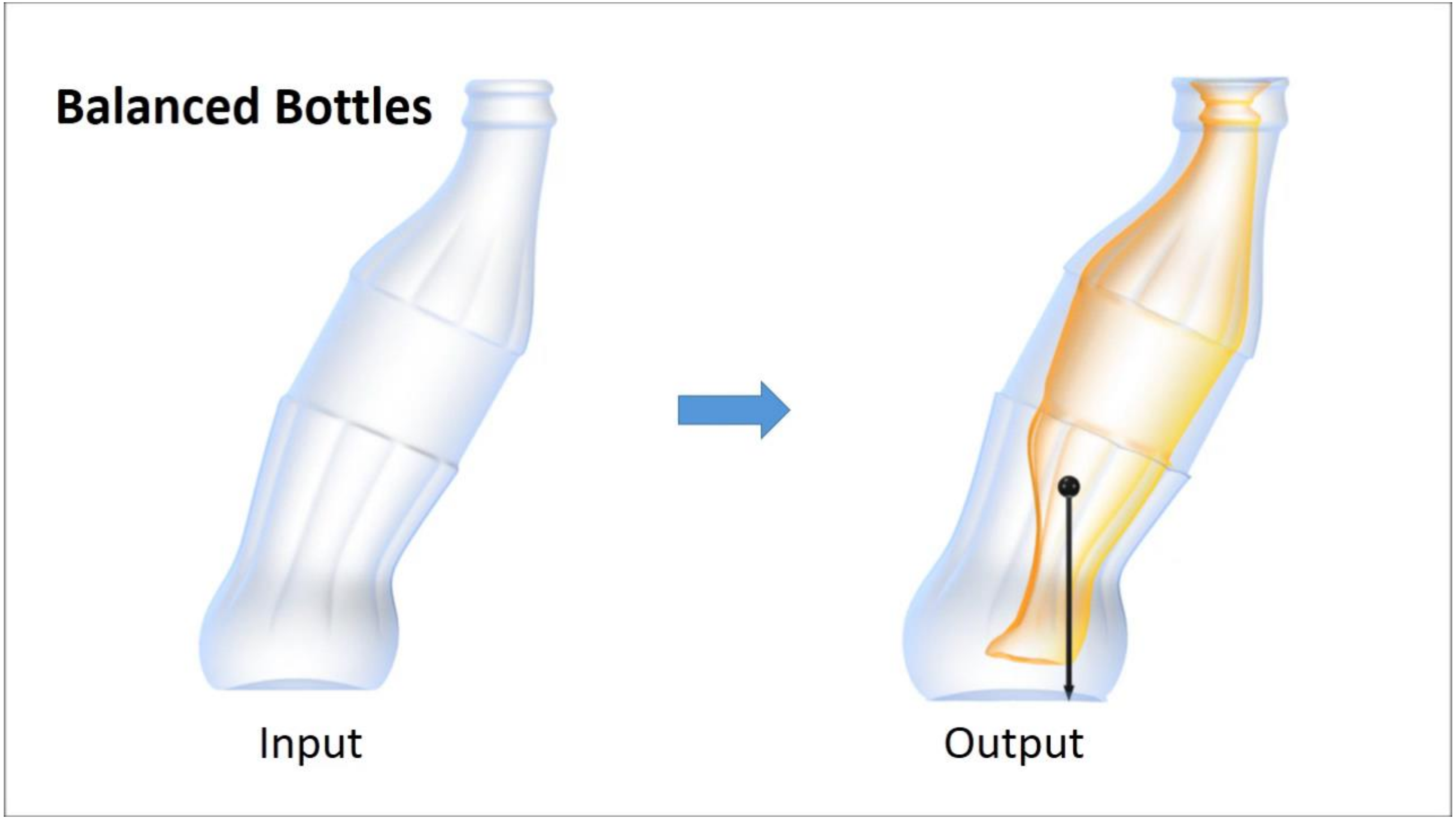
Spinning Turtle



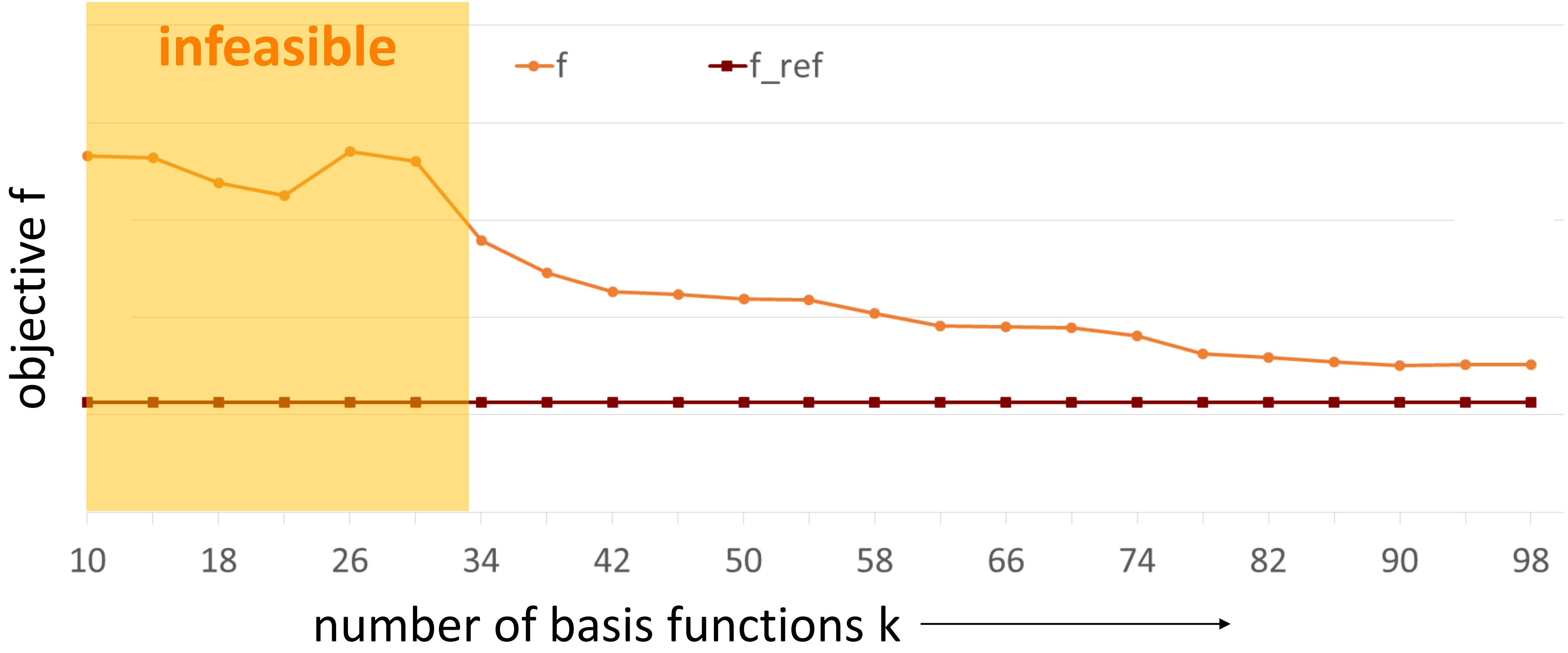
Rabbit Rolly-Polly



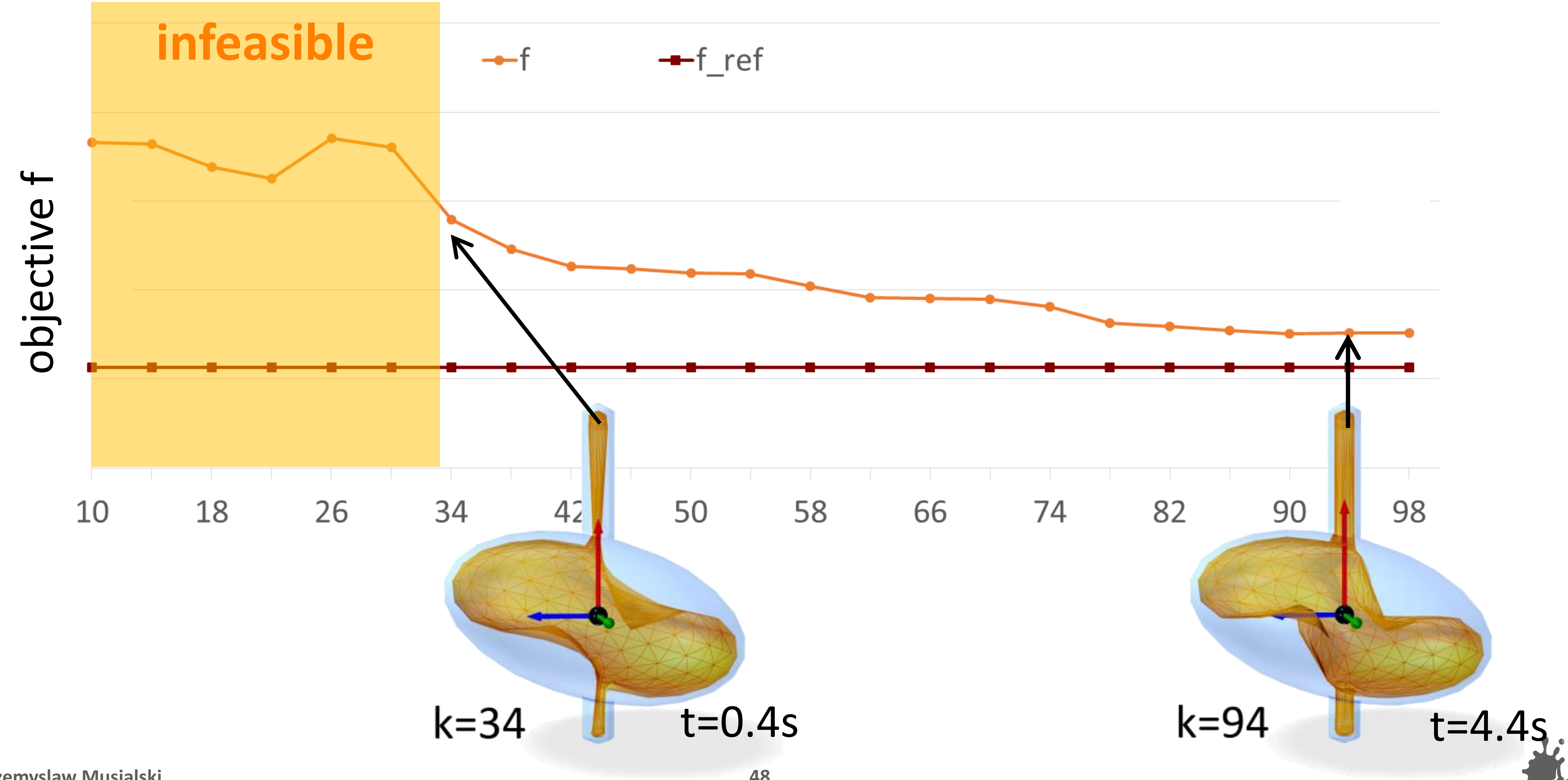
Balanced Bottles



Evaluation


infeasible
f
f_ref
objective f
10
18
26
34
42
50
58
66
74
82
90
98
number of basis functions k

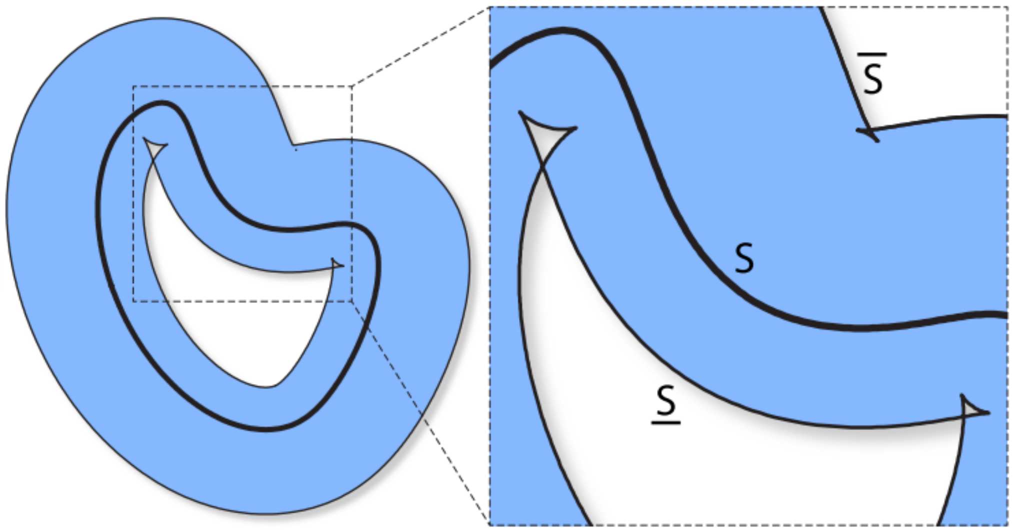
Evaluation



infeasible
f
f_ref
objective f
10
18
26
34
42
50
58
66
74
82
90
98
k=34
t=0.4s
k=94
t=4.4s

Discussion & Limitations

- skeleton dependence
  - our method relies on the skeleton
  - we use iterative mesh contraction (Mean Curvature Flow)



- design space limitation
  - we can only offset a surface up to the skeleton

Conclusions

- we proposed a novel framework for shape optimization
- we provide an elegant and efficient basis-reduction
- we demonstrate the method by optimizing mass properties for various goals

Thank you!