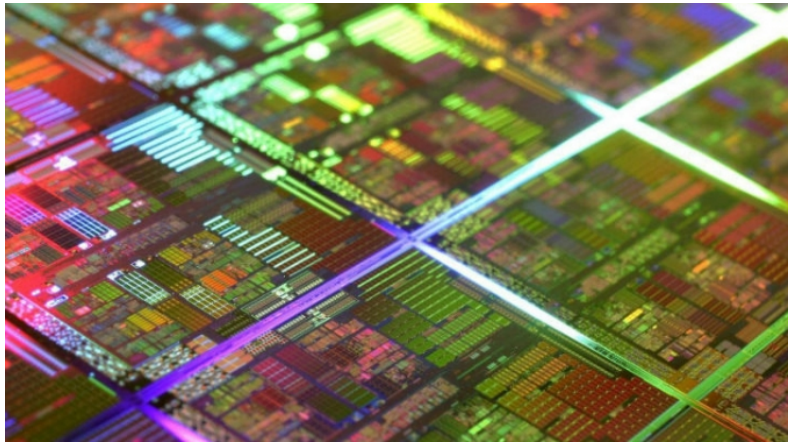
Constrained Deformation for Evolutionary Optimization

Daniel Sieger

Graphics & Geometry Group, Bielefeld University

Computer-aided Design and Engineering

Tools and practices for creating highly complex industrial products



Design Optimization and Shape Deformation

Design optimization

- Discover alternative designs with improved properties
- Decrease the need for costly real-world prototypes

Shape deformation

- Effective means to create design variations during optimization

Design Optimization and Shape Deformation

Design optimization

- Discover alternative designs with improved properties
- Decrease the need for costly real-world prototypes

Shape deformation

- Effective means to create design variations during optimization

Initial Design

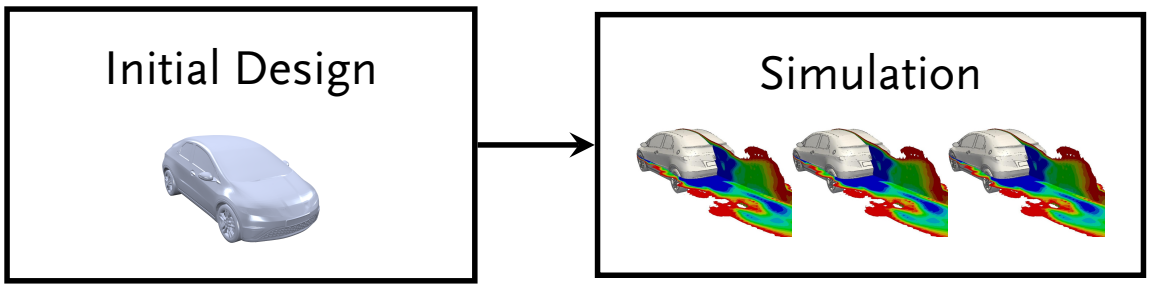
Design Optimization and Shape Deformation

Design optimization

- Discover alternative designs with improved properties
- Decrease the need for costly real-world prototypes

Shape deformation

- Effective means to create design variations during optimization



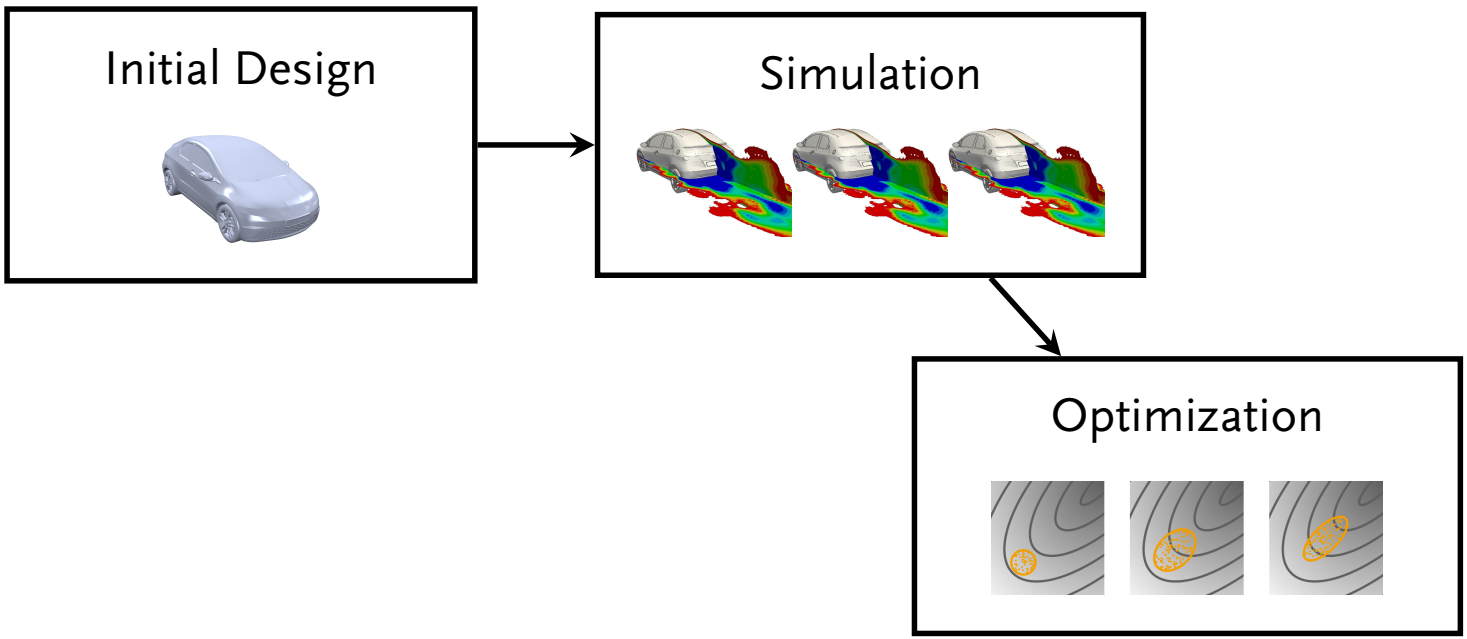
Design Optimization and Shape Deformation

Design optimization

- Discover alternative designs with improved properties
- Decrease the need for costly real-world prototypes

Shape deformation

- Effective means to create design variations during optimization



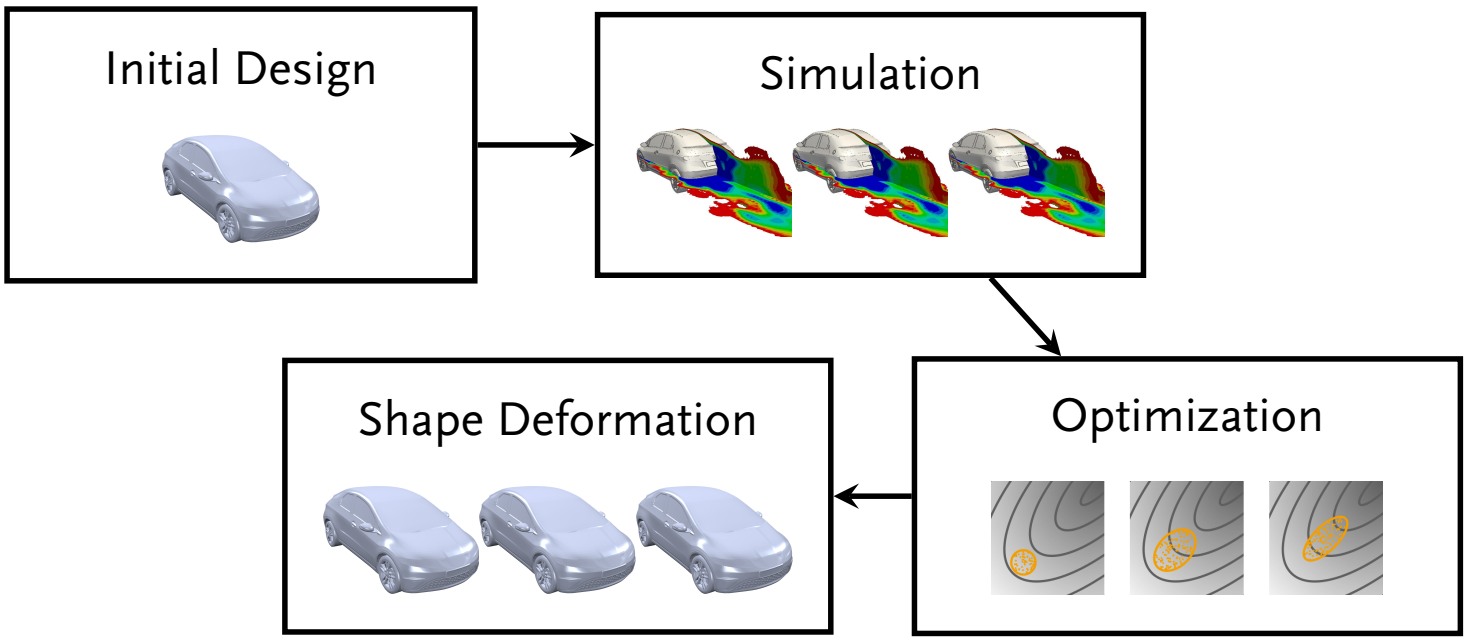
Design Optimization and Shape Deformation

Design optimization

- Discover alternative designs with improved properties
- Decrease the need for costly real-world prototypes

Shape deformation

- Effective means to create design variations during optimization



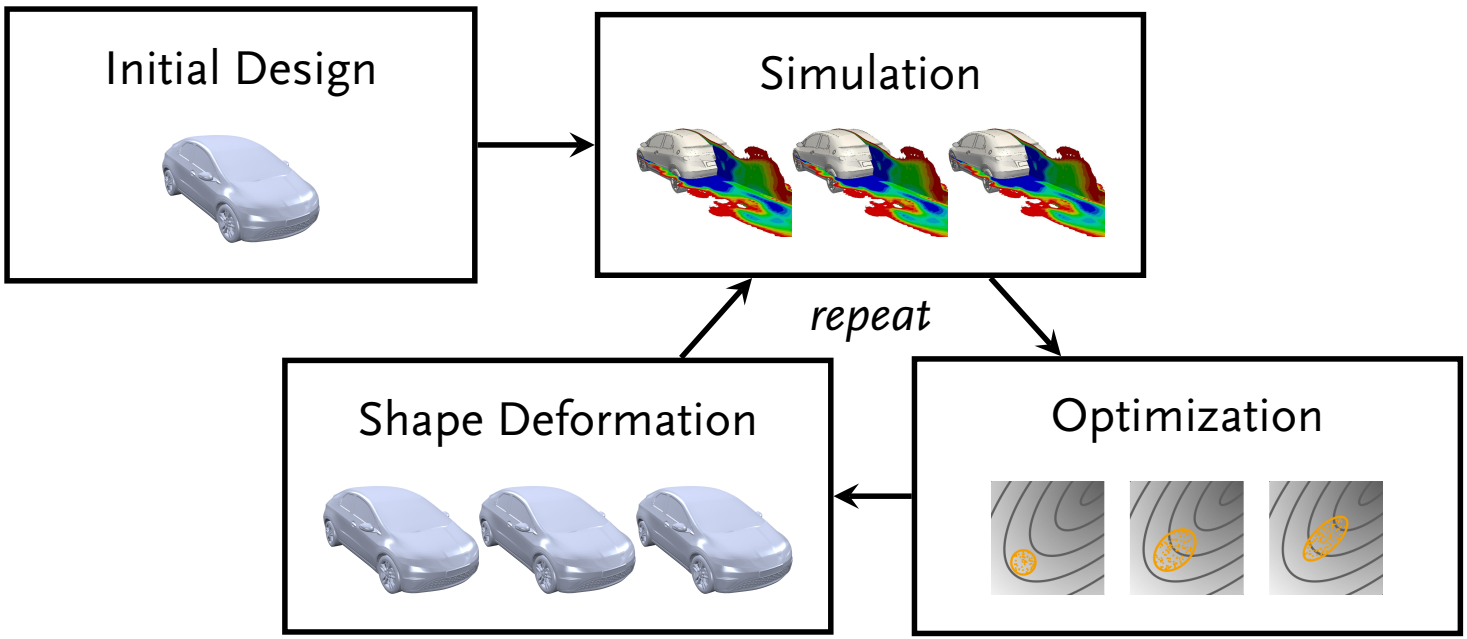
Design Optimization and Shape Deformation

Design optimization

- Discover alternative designs with improved properties
- Decrease the need for costly real-world prototypes

Shape deformation

- Effective means to create design variations during optimization



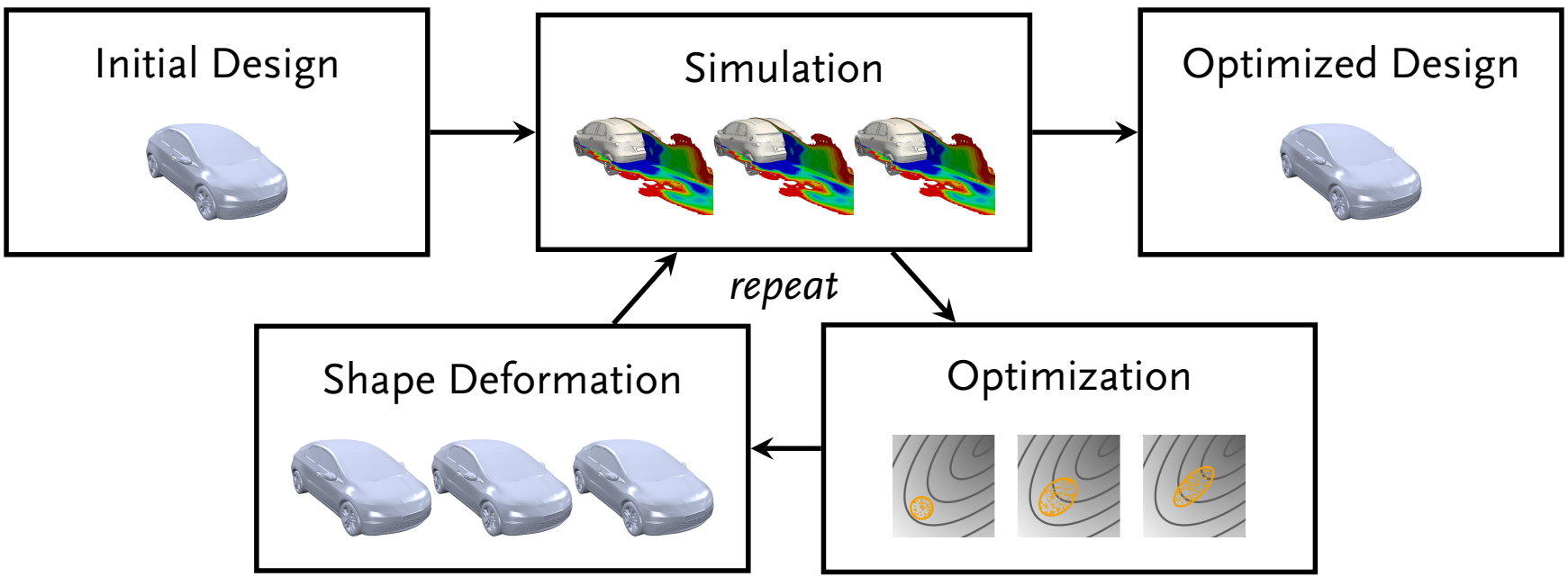
Design Optimization and Shape Deformation

Design optimization

- Discover alternative designs with improved properties
- Decrease the need for costly real-world prototypes

Shape deformation

- Effective means to create design variations during optimization



Key Questions

1. What is a *good* deformation technique for design optimization?
    - Fundamental requirements and properties
    - Select from the wide variety of methods

Key Questions

1. What is a *good* deformation technique for design optimization?
   - Fundamental requirements and properties
   - Select from the wide variety of methods
2. How to *apply* and *improve* existing techniques?
   - Effective application within design optimization
   - Address specific challenges

Key Questions

1. What is a *good* deformation technique for design optimization?
   - Fundamental requirements and properties
   - Select from the wide variety of methods
2. How to *apply* and *improve* existing techniques?
   - Effective application within design optimization
   - Address specific challenges
3. How to incorporate *geometric constraints* into the deformation?
   - Maintain critical properties such as planarity
   - Maintain deformation quality and flexibility

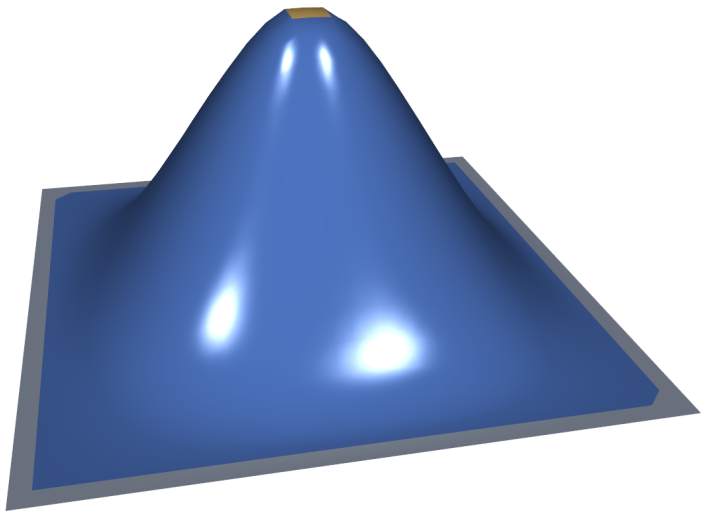
Outline

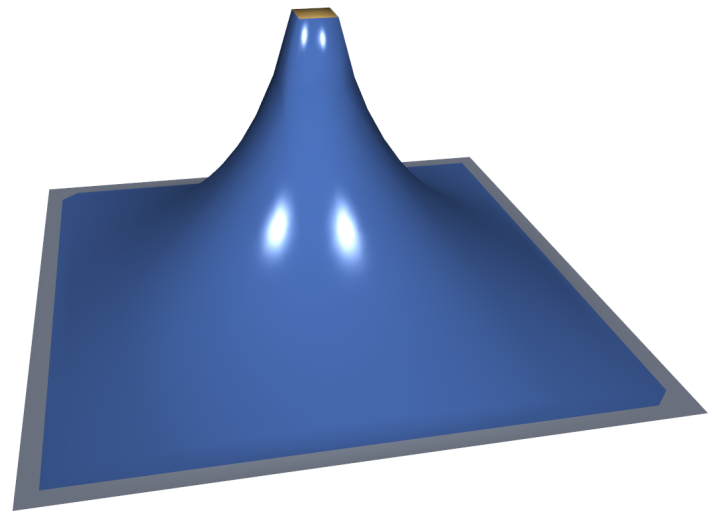
1. Shape deformation methods for design optimization
   - Introduce the state of the art
   - Benchmarks and comparisons
2. Advanced RBF deformation techniques
   - Automatic CAD-based design optimization
   - Avoid costly remeshing
3. Constrained deformation
   - Effective constraint preservation
   - High-quality, flexible deformations

Shape Deformation for Design Optimization

Thin Shell Deformation [1]

- Physically-inspired technique suitable for sheet metal surfaces
- Flexible modeling of material behavior
- Based on the minimization of stretching and bending energies

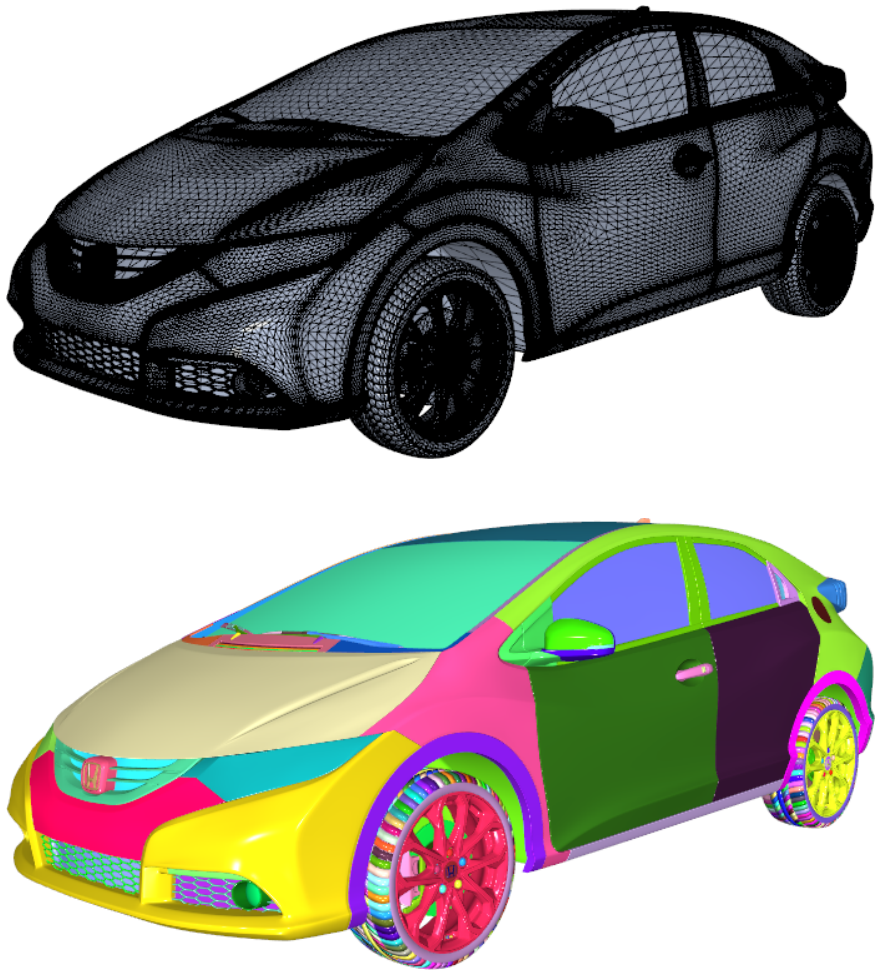




1. Botsch et al., *On Linear Variational Surface Deformation Methods*, Trans. on Visualization and Computer Graphics, 2008

Limitations

Assumption: Surface $\mathcal{S}$ is a proper triangle mesh



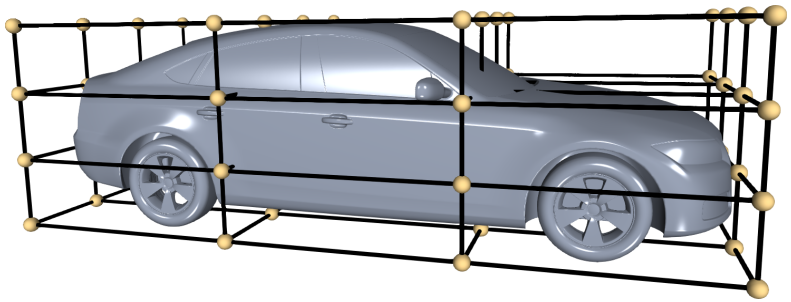
Space Deformation Methods

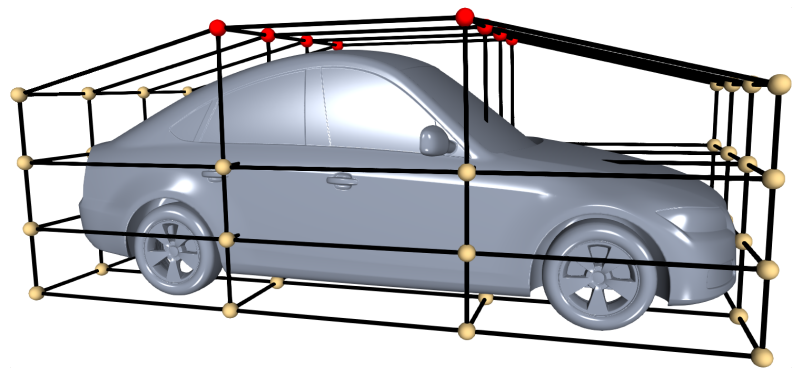
- Instead of deforming the surface $\mathcal{S}$, deform embedding space Ω
  - + Representation independence
  - + Disconnected components
  - + Robustness against defects
- Construct a space deformation function $\boldsymbol{d}: \Omega \subset \mathbb{R}^3 \rightarrow \mathbb{R}^3$

Free-Form Deformation[2] (FFD)

- Idea: Embed object in control grid and deform grid
- Procedure:
  1. Compute local coordinates within the control grid
  2. Select and move control points
  3. Deform object according to updated control points

$$\boldsymbol{d}(\boldsymbol{x}) = \sum_{i=0}^{l} \sum_{j=0}^{m} \sum_{k=0}^{n} \delta \boldsymbol{c}_{ijk} \varphi_i(u_1) \varphi_j(u_2) \varphi_k(u_3),$$

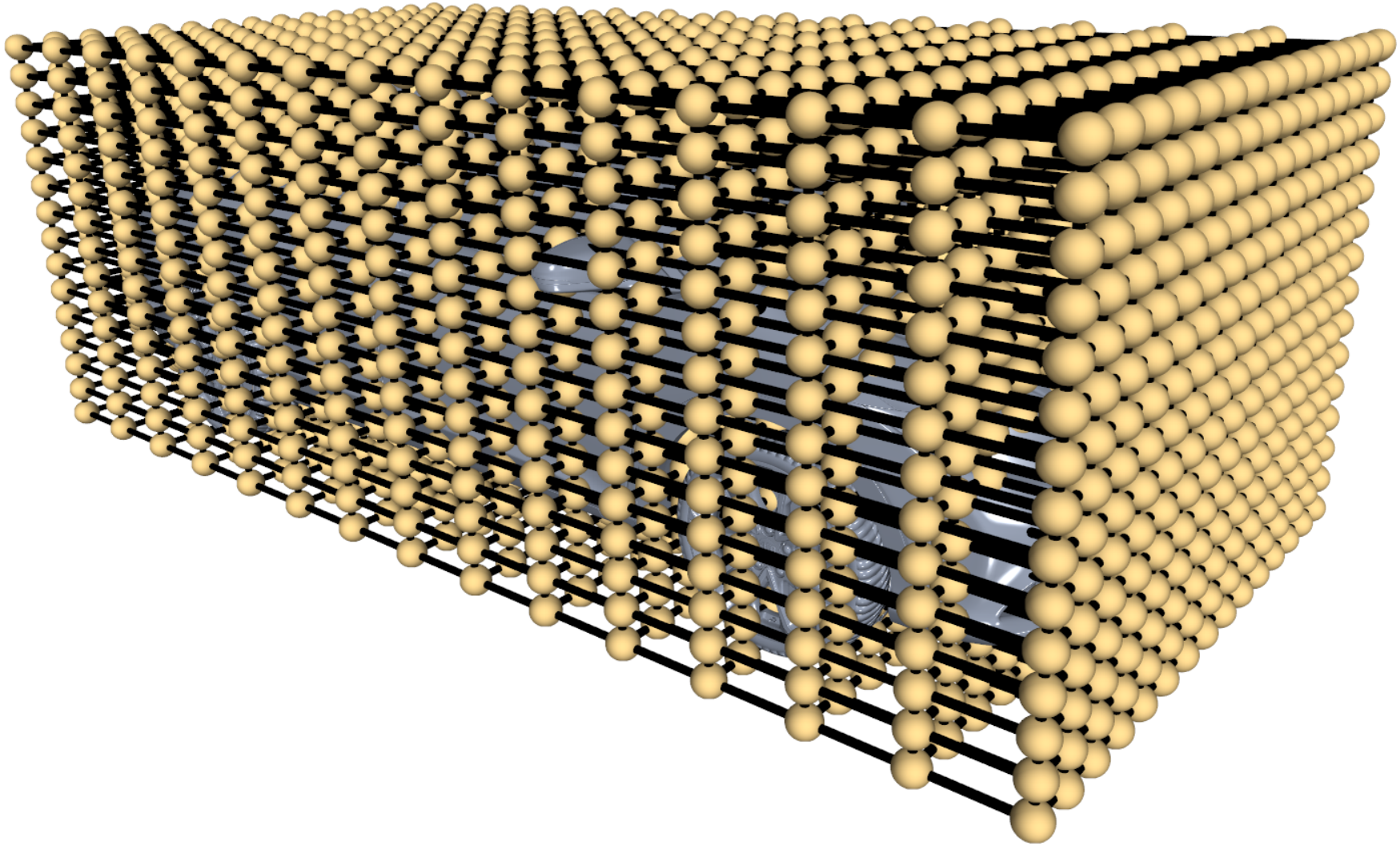




2. Sederberg and Parry, *Free-Form Deformation of Solid Geometric Models*, SIGGRAPH 1986

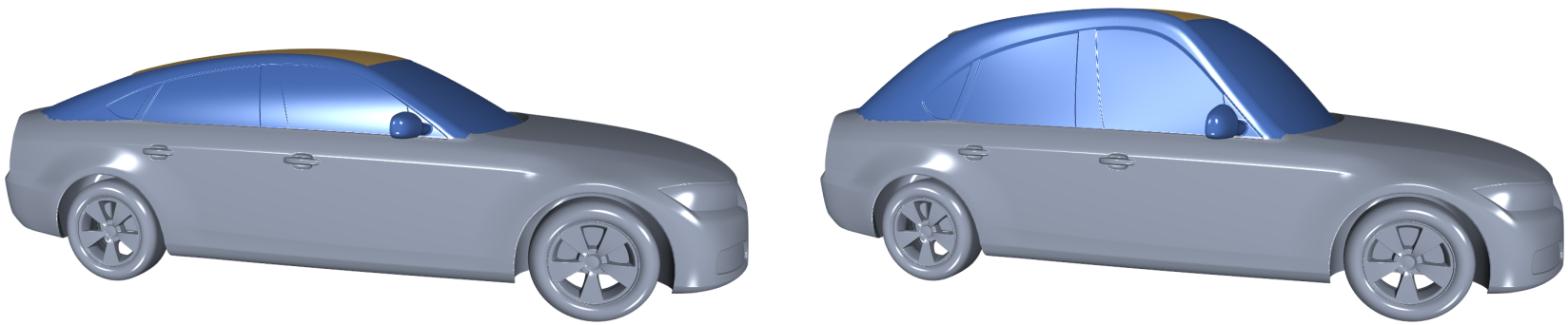
FFD Limitations

Manipulating large control grids by hand is tedious



Direct Manipulation FFD[3]

- Move object points directly (constraints)
- Compute control point displacements satisfying constraints
- Requires solving a linear system (pseudo-inverse, SVD)



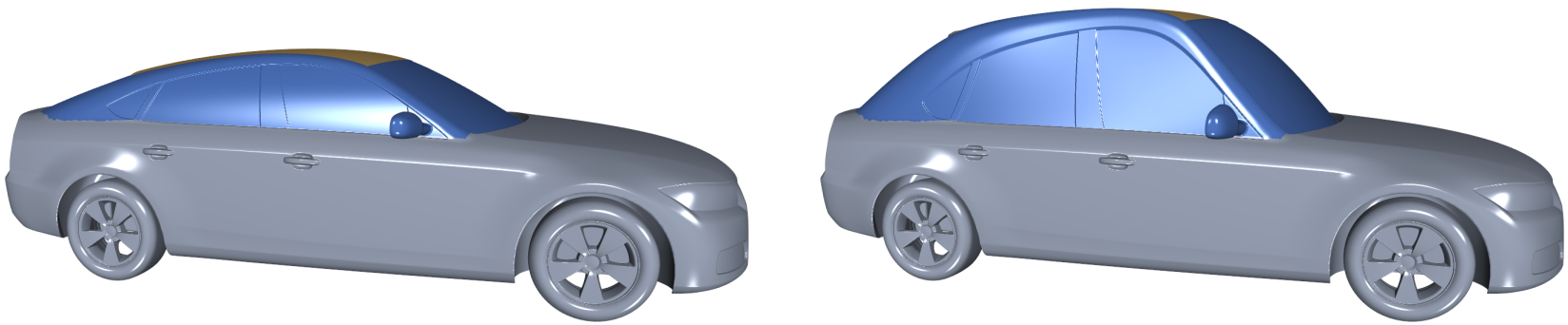
3. Hsu et al. *Direct Manipulation of Free-Form Deformations*, SIGGRAPH 1992

Direct Manipulation FFD[3]

- Move object points directly (constraints)
- Compute control point displacements satisfying constraints
- Requires solving a linear system (pseudo-inverse, SVD)

– Minimizes constraint error and control point movement

– Not physically plausible, not precise enough



3. Hsu et al. *Direct Manipulation of Free-Form Deformations*, SIGGRAPH 1992

Radial Basis Function Deformation[4]

- Deformation as scattered data interpolation problem
  - *Exactly* interpolate prescribed displacements
  - *Smoothly* interpolate displacements through space

4. Botsch and Kobbelt, *Real-Time Shape Editing Using Radial Basis Functions*, Computer Graphics Forum, 2005

Radial Basis Function Deformation[4]

- Deformation as scattered data interpolation problem
  - *Exactly* interpolate prescribed displacements
  - *Smoothly* interpolate displacements through space

➔ Radial basis functions (RBFs)

4. Botsch and Kobbelt, *Real-Time Shape Editing Using Radial Basis Functions*, Computer Graphics Forum, 2005

Radial Basis Function Deformation[4]

- Deformation as scattered data interpolation problem
  - *Exactly* interpolate prescribed displacements
  - *Smoothly* interpolate displacements through space

➔ Radial basis functions (RBFs)

$$\boldsymbol{d}\colon \mathbb{R}^3 \to \mathbb{R}^3$$

4. Botsch and Kobbelt, *Real-Time Shape Editing Using Radial Basis Functions*, Computer Graphics Forum, 2005

Radial Basis Function Deformation[4]

- Deformation as scattered data interpolation problem
  - *Exactly* interpolate prescribed displacements
  - *Smoothly* interpolate displacements through space

➔ Radial basis functions (RBFs)

$$\boldsymbol{d}(\boldsymbol{x}) = \sum_{j=1}^{m} \boldsymbol{w}_j \varphi_j(\boldsymbol{x}) + \boldsymbol{\pi}(\boldsymbol{x})$$

4. Botsch and Kobbelt, *Real-Time Shape Editing Using Radial Basis Functions*, Computer Graphics Forum, 2005

Radial Basis Function Deformation[4]

- Deformation as scattered data interpolation problem
  - *Exactly* interpolate prescribed displacements
  - *Smoothly* interpolate displacements through space

➔ Radial basis functions (RBFs)

basis functions at centers $\boldsymbol{c}_j$

$$\boldsymbol{d}(\boldsymbol{x}) = \sum_{j=1}^{m} \boldsymbol{w}_j \varphi_j(\boldsymbol{x}) + \boldsymbol{\pi}(\boldsymbol{x})$$

4. Botsch and Kobbelt, *Real-Time Shape Editing Using Radial Basis Functions*, Computer Graphics Forum, 2005

Radial Basis Function Deformation[4]

- Deformation as scattered data interpolation problem
  - *Exactly* interpolate prescribed displacements
  - *Smoothly* interpolate displacements through space

➔ Radial basis functions (RBFs)

basis functions at centers $\boldsymbol{c}_j$

$$\boldsymbol{d}(\boldsymbol{x}) = \sum_{j=1}^{m} \boldsymbol{w}_j \varphi_j(\boldsymbol{x}) + \boldsymbol{\pi}(\boldsymbol{x})$$

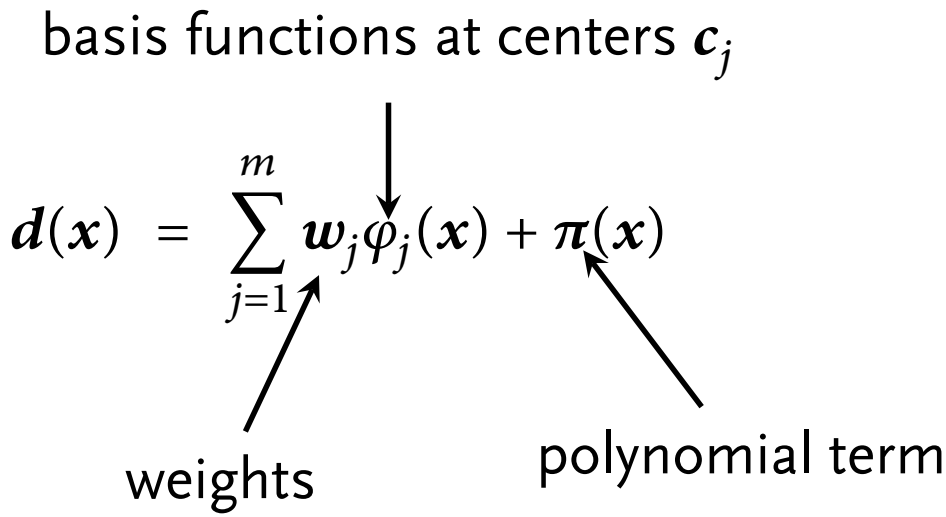
weights

4. Botsch and Kobbelt, *Real-Time Shape Editing Using Radial Basis Functions*, Computer Graphics Forum, 2005

Radial Basis Function Deformation[4]

- Deformation as scattered data interpolation problem
  - *Exactly* interpolate prescribed displacements
  - *Smoothly* interpolate displacements through space

➔ Radial basis functions (RBFs)



4. Botsch and Kobbelt, *Real-Time Shape Editing Using Radial Basis Functions*, Computer Graphics Forum, 2005

Radial Basis Functions

- Various choices: Gaussian, multiquadrics, thin plate spline...
- Choose triharmonic basis functions $\varphi(r) = r^3$ so that $\boldsymbol{d}$ minimizes fairness energy:

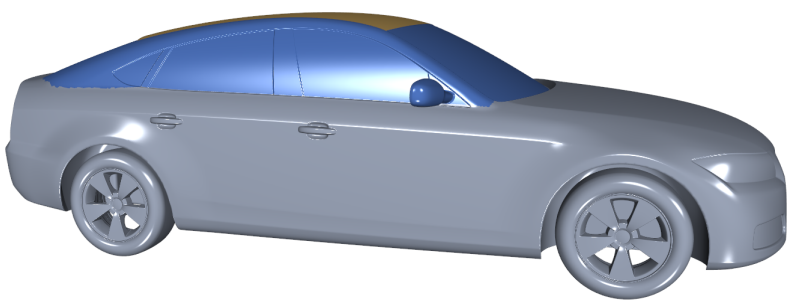
$$\int_{\mathbb{R}^3} \left\| \frac{\partial^3 \boldsymbol{d}}{\partial x^3} \right\|^2 + \left\| \frac{\partial^3 \boldsymbol{d}}{\partial x^2 \partial y} \right\|^2 + \ldots + \left\| \frac{\partial^3 \boldsymbol{d}}{\partial z^3} \right\|^2 \, \mathrm{dV}$$

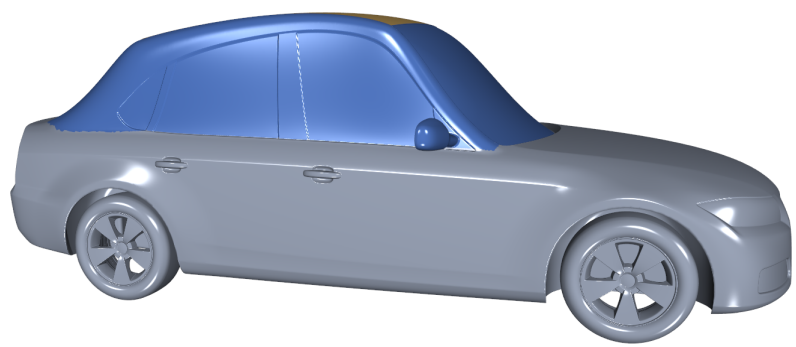
Radial Basis Functions

- Various choices: Gaussian, multiquadrics, thin plate spline...
- Choose triharmonic basis functions $\varphi(r) = r^3$ so that $\boldsymbol{d}$ minimizes fairness energy:

$$\int_{\mathbb{R}^3} \left\| \frac{\partial^3 \boldsymbol{d}}{\partial x^3} \right\|^2 + \left\| \frac{\partial^3 \boldsymbol{d}}{\partial x^2 \partial y} \right\|^2 + \ldots + \left\| \frac{\partial^3 \boldsymbol{d}}{\partial z^3} \right\|^2 \mathrm{dV}$$

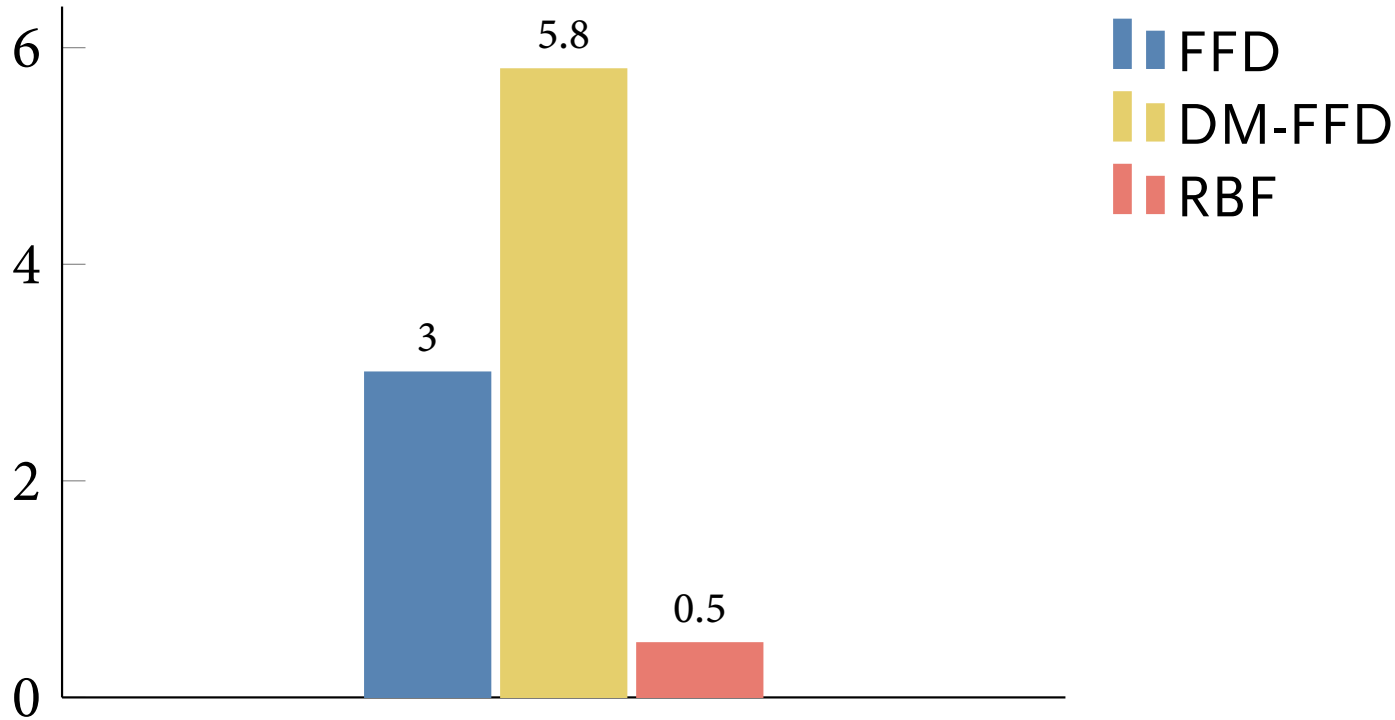
- Determine weights and polynomial coefficients
  ➔ Solve linear system





Performance Comparison[5]

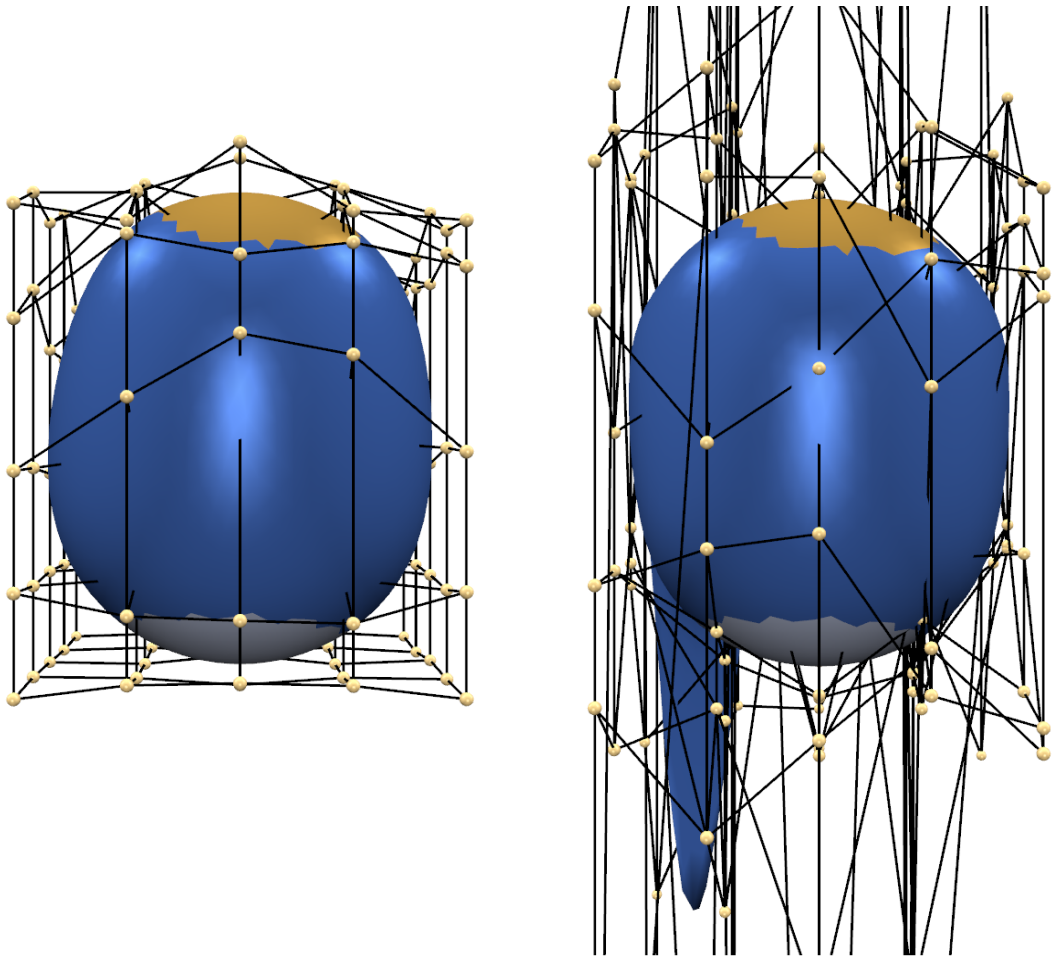
Precomputed deformation, times in seconds



5. Sieger et al., *On Shape Deformation Techniques for Simulation-based Design Optimization*, New Challenges in Grid Generation and Adaptivity for Scientific Computing, 2015

Robustness

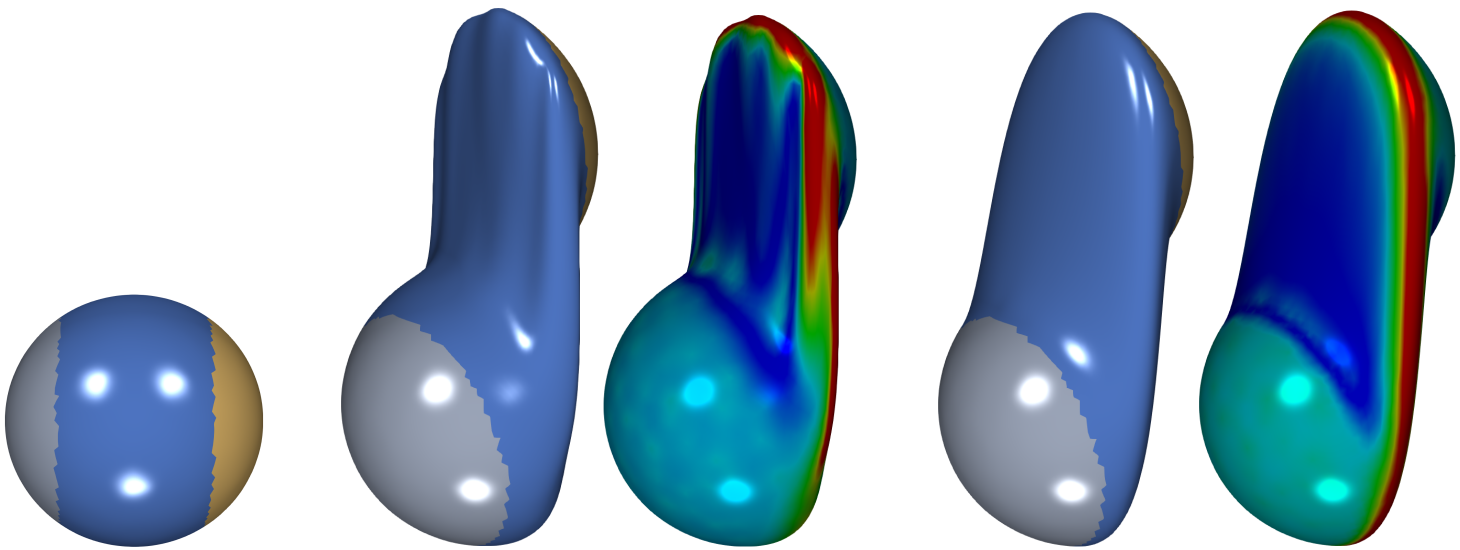
DM-FFD: Solving the linear system using SVD requires clamping of small singular values. Since the clamping value is not known in advance, robustness artifacts might occur.



Quality

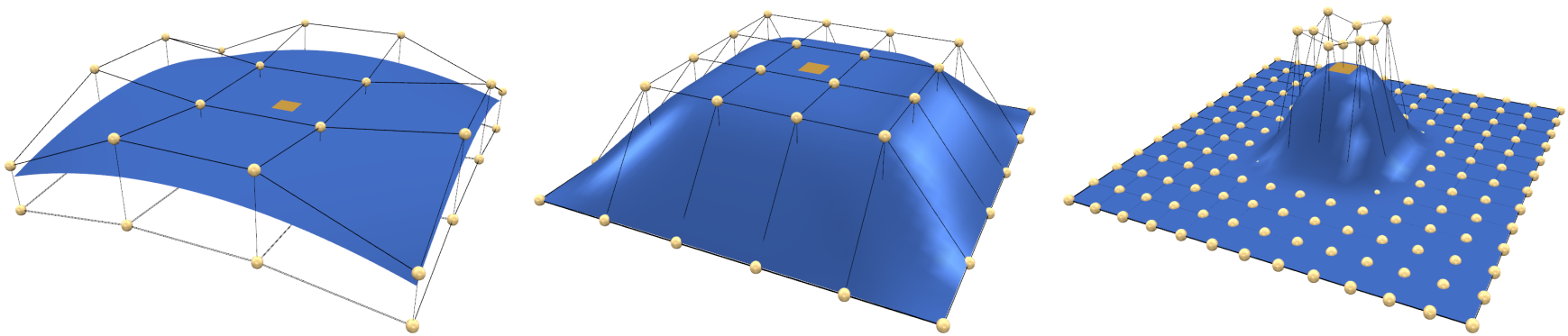
Quality comparison using mean curvature:

- FFD methods (center) might lead to smoothness artifacts
- RBFs (right) yield highly smooth results



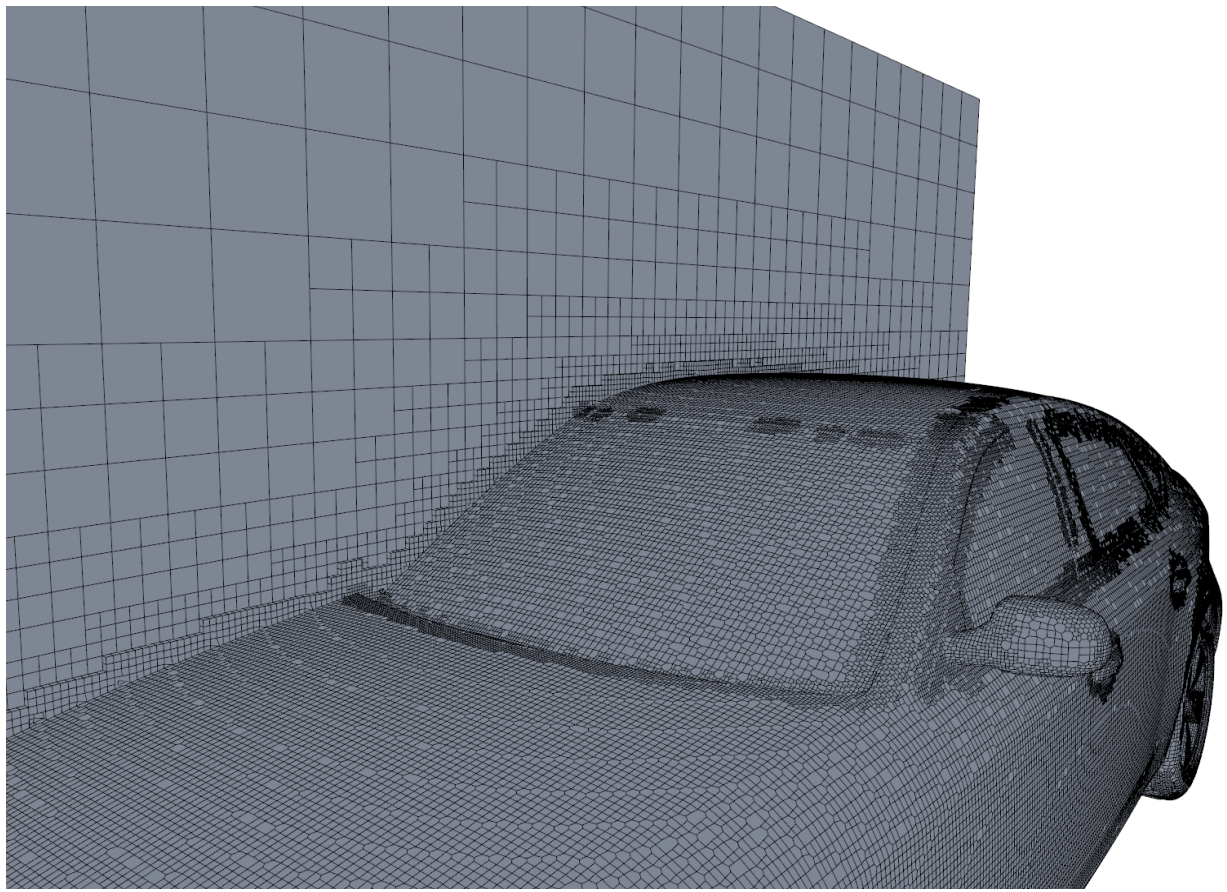
Quality

FFD: The shape of the deformation depends on grid resolution and not on some form of physically-inspired energy minimization



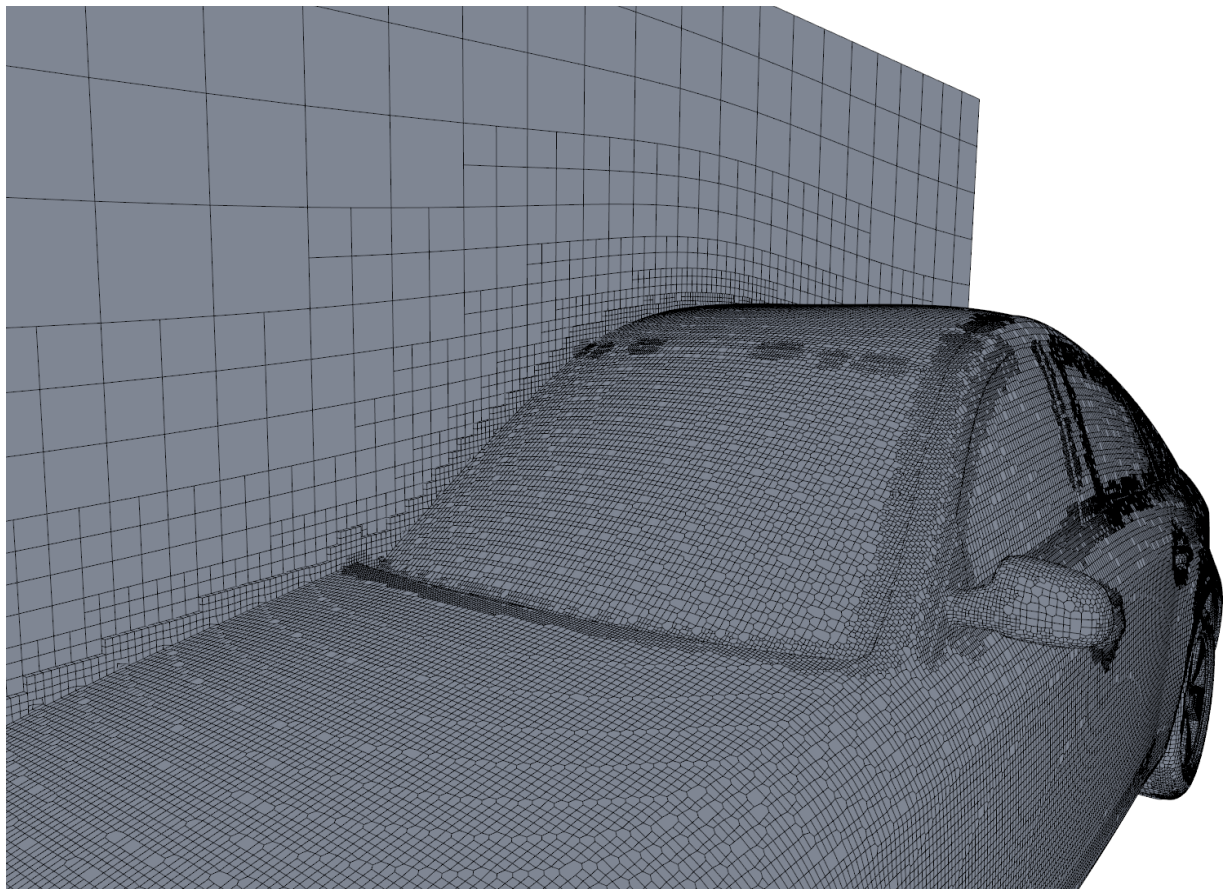
Volume Mesh Deformation

- Deformation of a hex-dominant volume mesh for CFD simulation



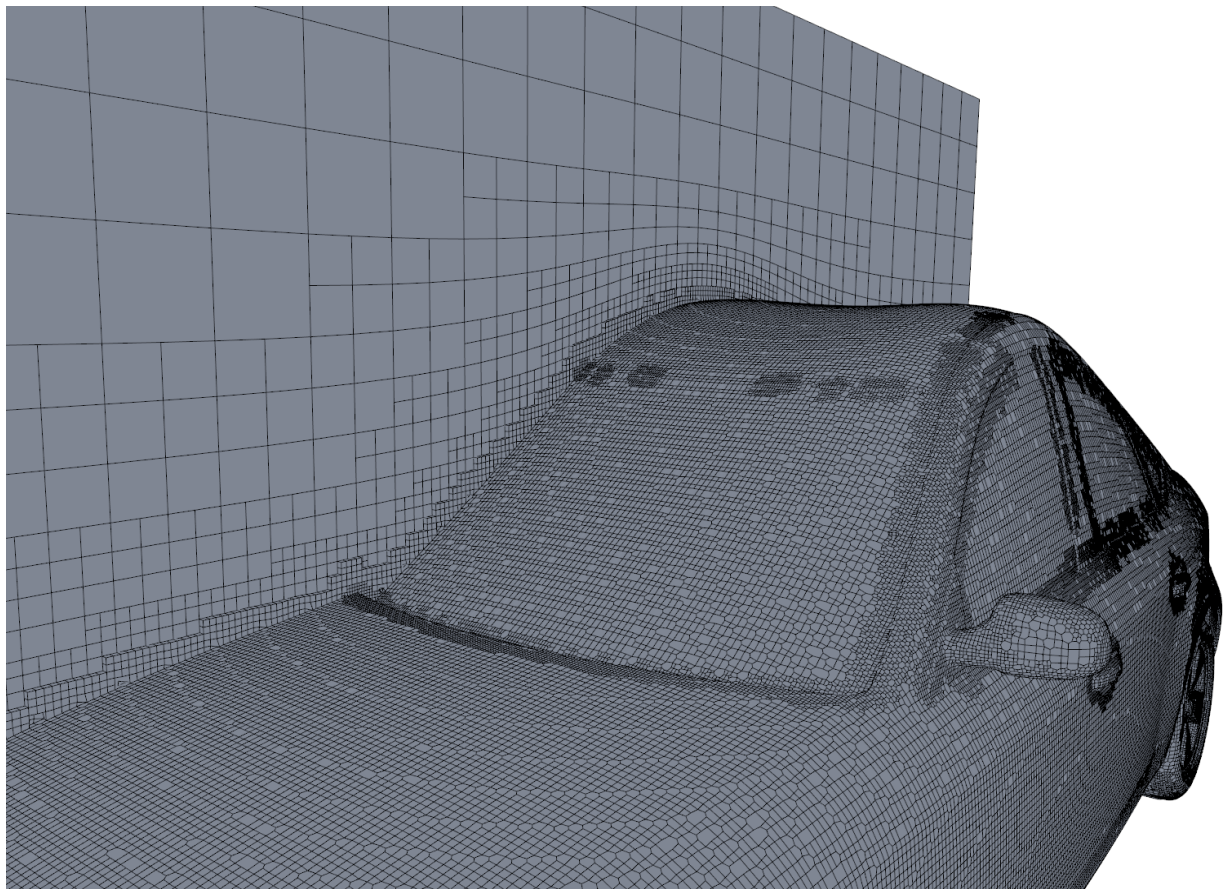
Volume Mesh Deformation

- Deformation of a hex-dominant volume mesh for CFD simulation



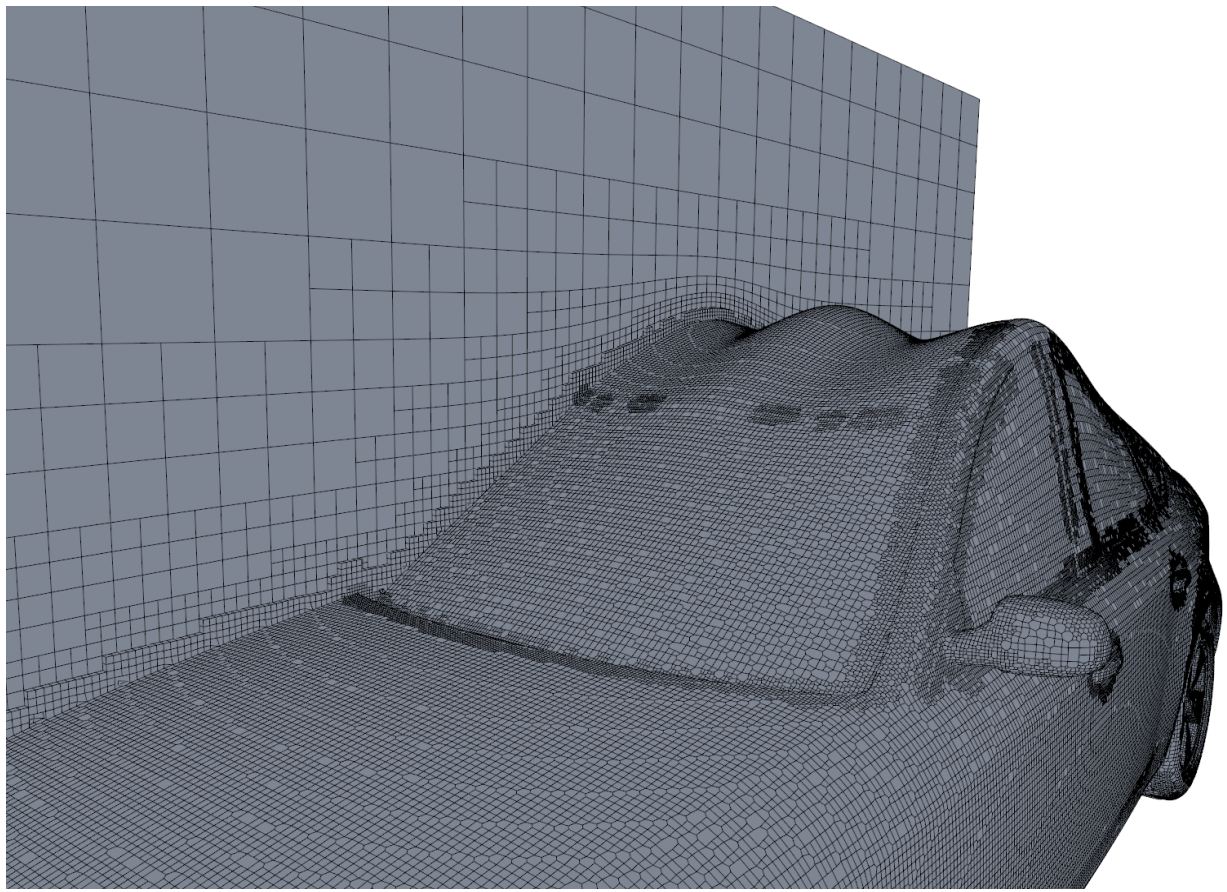
Volume Mesh Deformation

- Deformation of a hex-dominant volume mesh for CFD simulation



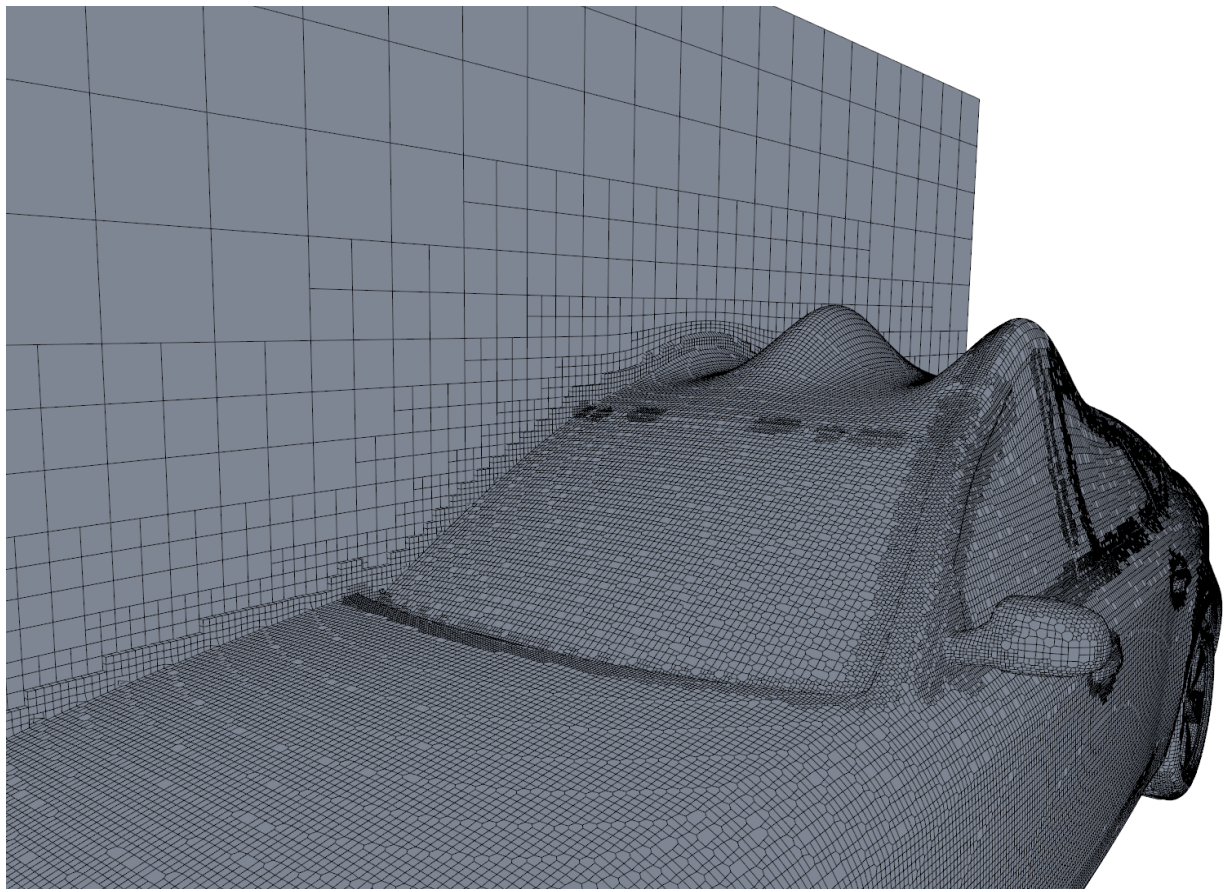
Volume Mesh Deformation

- Deformation of a hex-dominant volume mesh for CFD simulation



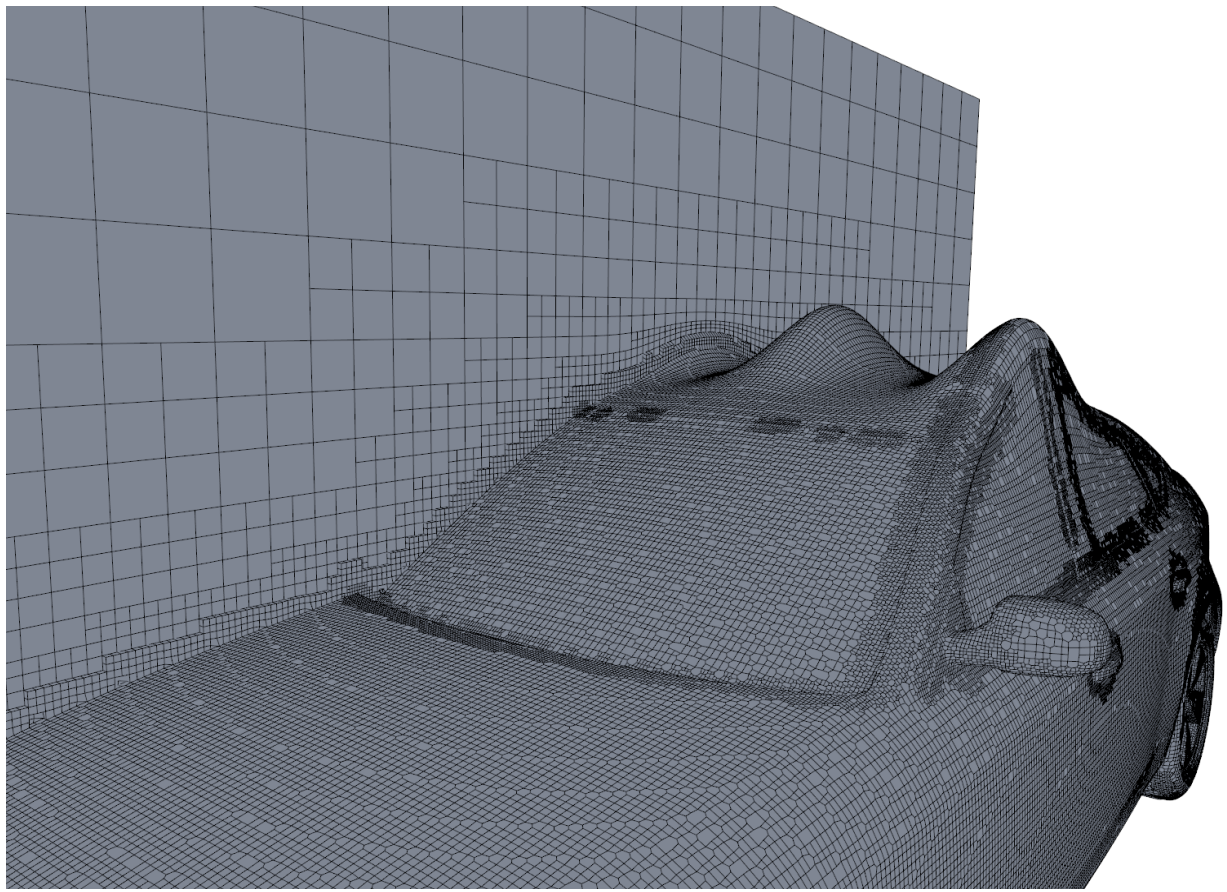
Volume Mesh Deformation

- Deformation of a hex-dominant volume mesh for CFD simulation



Volume Mesh Deformation

- Deformation of a hex-dominant volume mesh for CFD simulation
- Depending on the FFD grid resolution the mesh becomes unusable for simulation



Evaluation Summary

	Performance	Robustness	Quality	Adaptivity	Precision
FFD	•	+	•	−	−
DM-FFD	−	•	•	−	•
RBF	•	+	+	+	+

• : neutral + : positive − : negative

Key Questions and Answers

1. What is a *good* deformation technique for design optimization?

   Kernel-based space deformations provide the quality, flexibility, and robustness required for design optimization.

Key Questions and Answers

1. What is a *good* deformation technique for design optimization?

   Kernel-based space deformations provide the quality, flexibility, and robustness required for design optimization.

2. How to *apply* and *improve* existing techniques?

Advanced RBF Deformation Techniques

Advanced RBF Deformation Techniques [6]

- Unified framework for combined surface and volume deformation
- Preventing self-intersections by splitting
- Performance improvements through advanced linear solvers

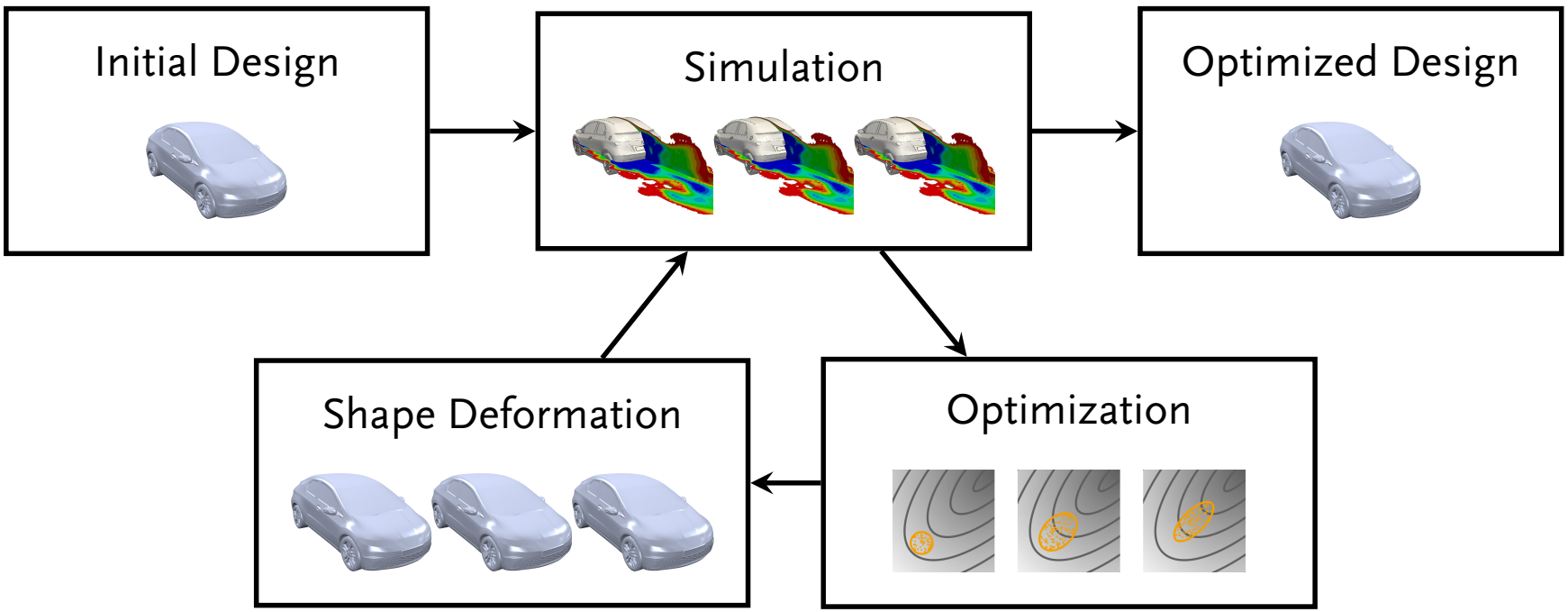
6. Sieger et al., *RBF Morphing Techniques for Simulation-based Design Optimization*, Engineering with Computers, 2014

Advanced RBF Deformation Techniques [6]

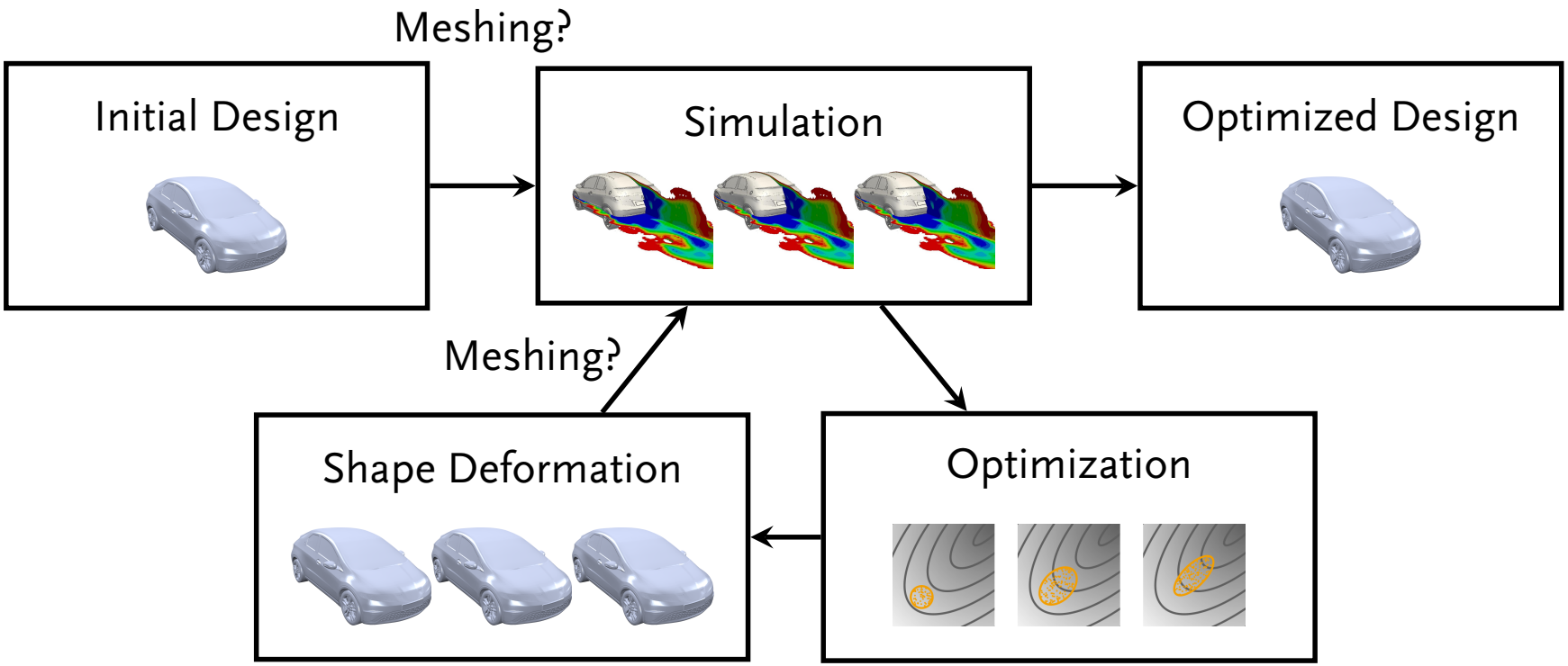
- Unified framework for combined surface and volume deformation
- Preventing self-intersections by splitting
- Performance improvements through advanced linear solvers

6. Sieger et al., *RBF Morphing Techniques for Simulation-based Design Optimization*, Engineering with Computers, 2014

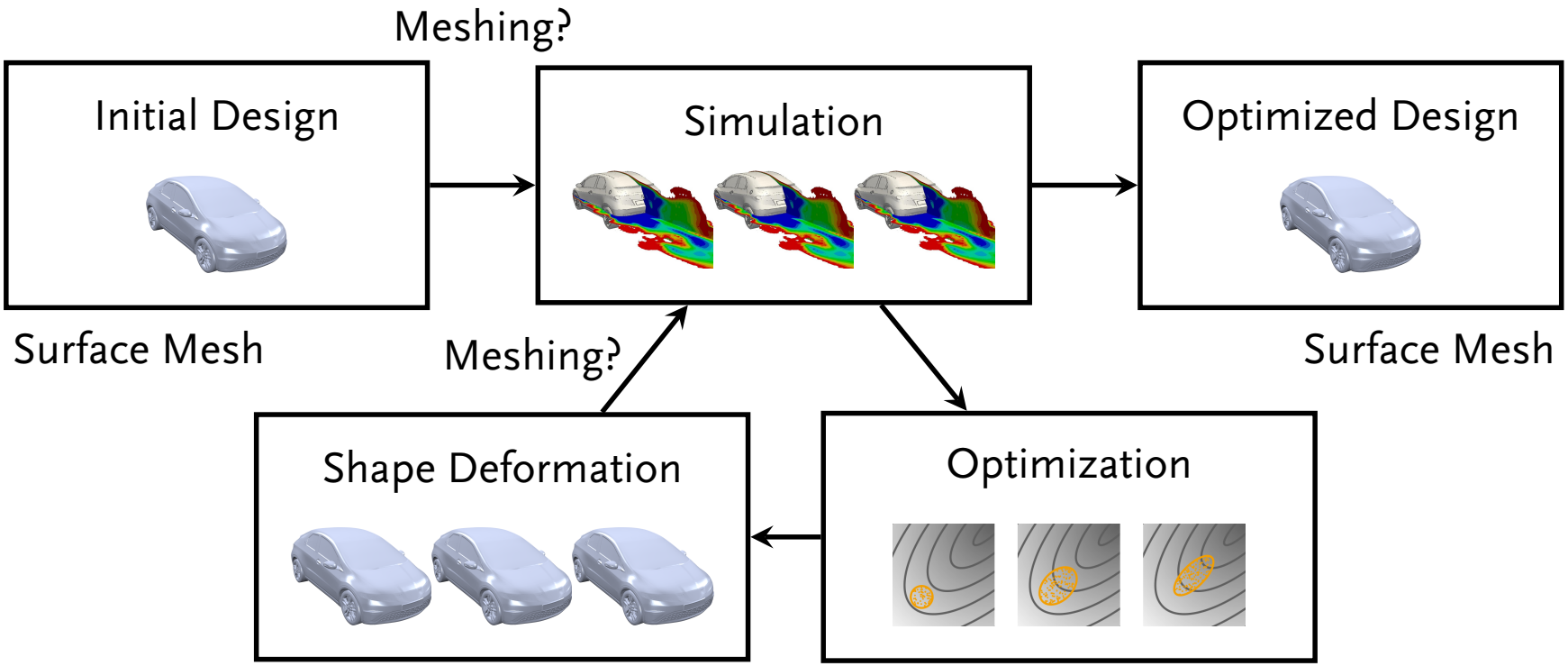
Shape Deformation & Design Optimization



Shape Deformation & Design Optimization

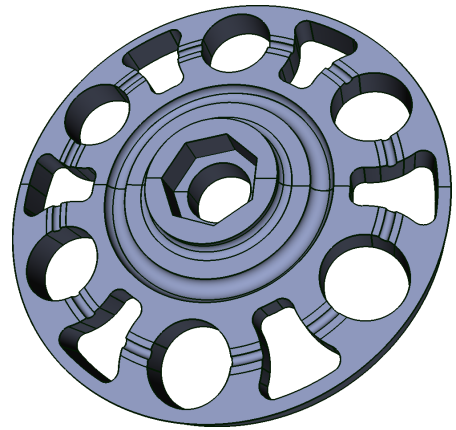


Shape Deformation & Design Optimization

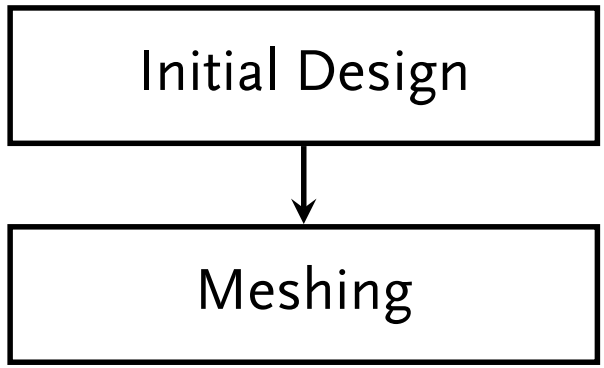


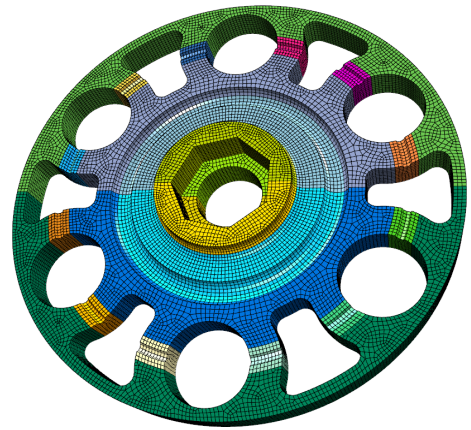
Generalized Design Optimization Scenario

Initial Design

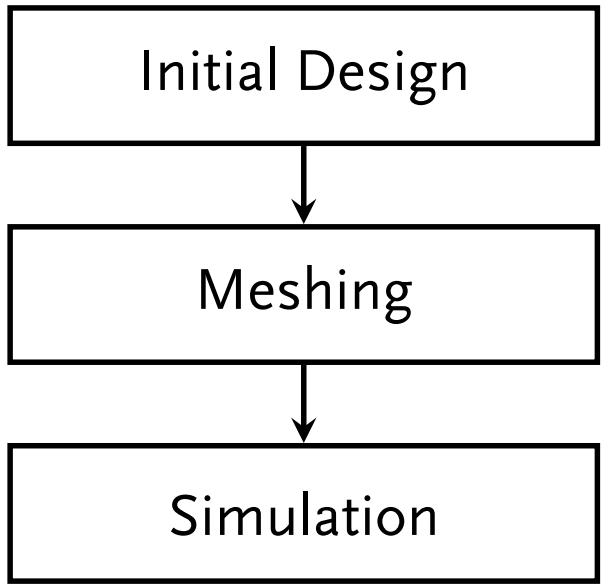


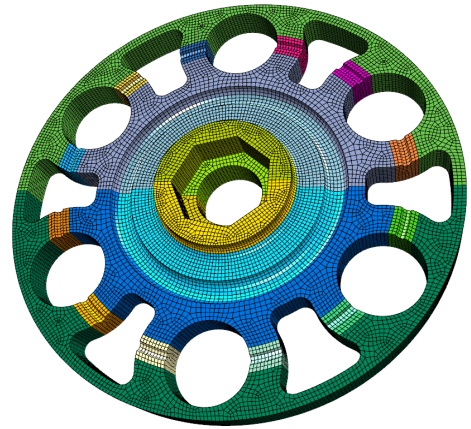
Generalized Design Optimization Scenario



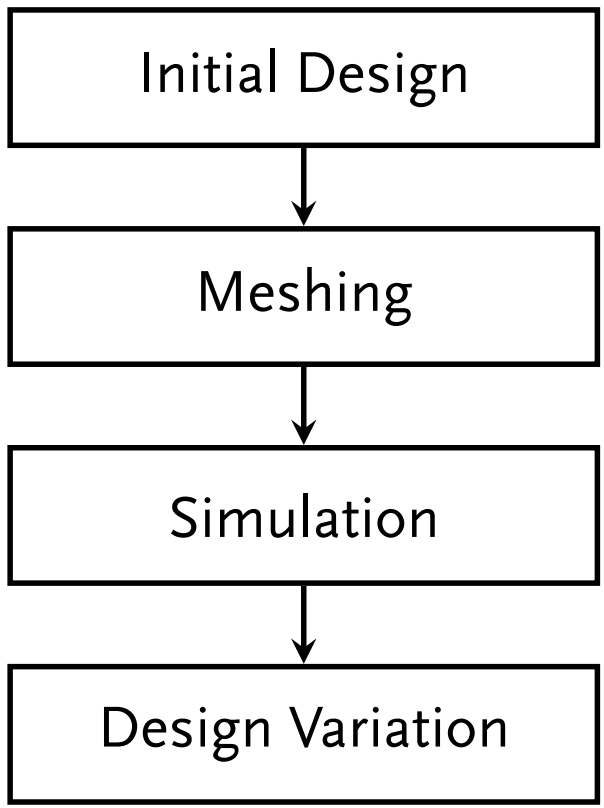


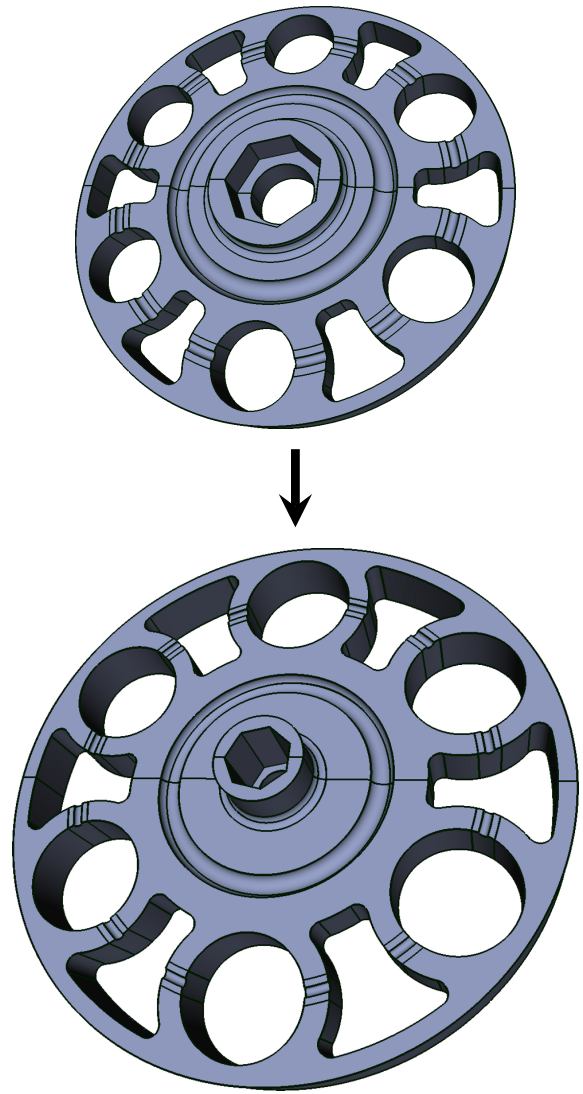
Generalized Design Optimization Scenario



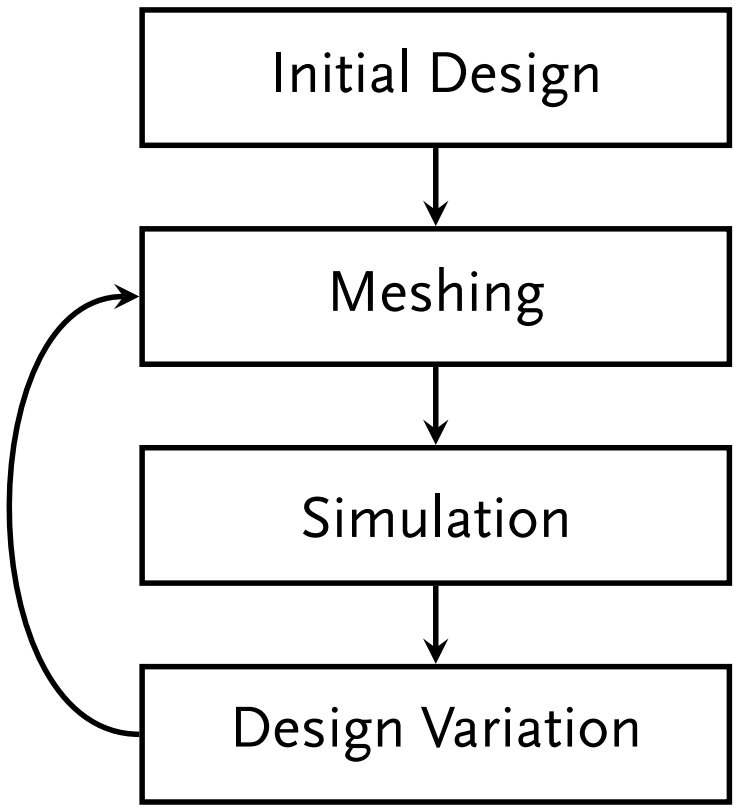


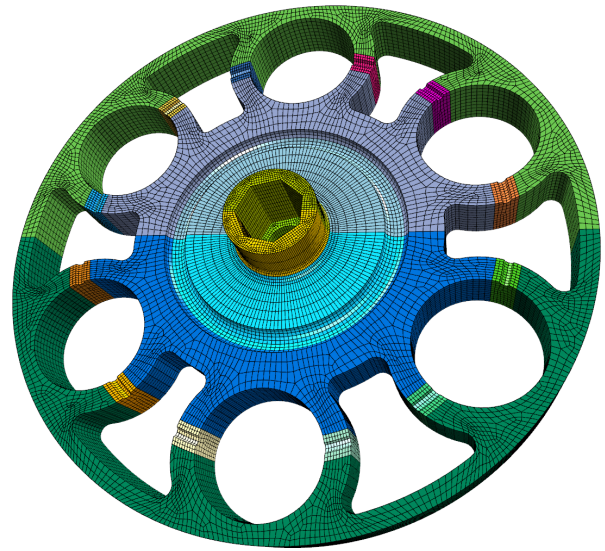
Generalized Design Optimization Scenario



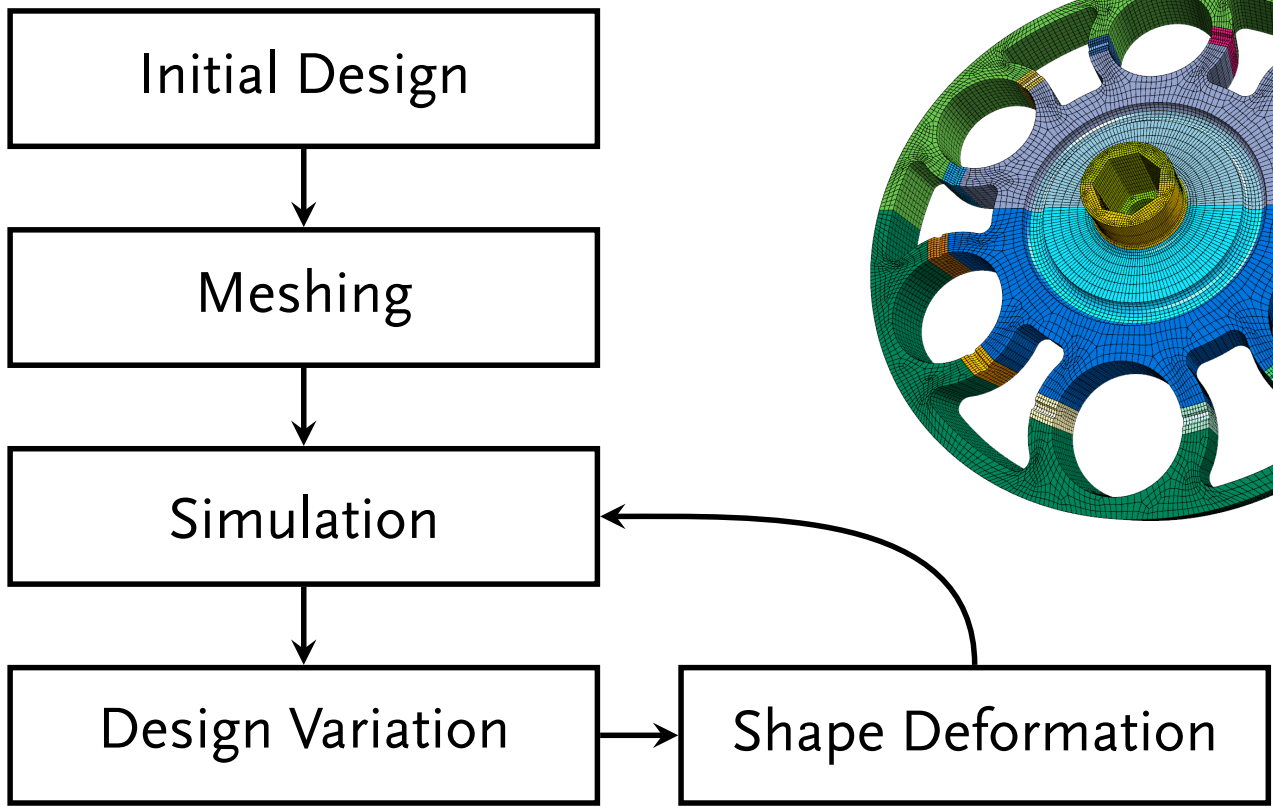


Generalized Design Optimization Scenario

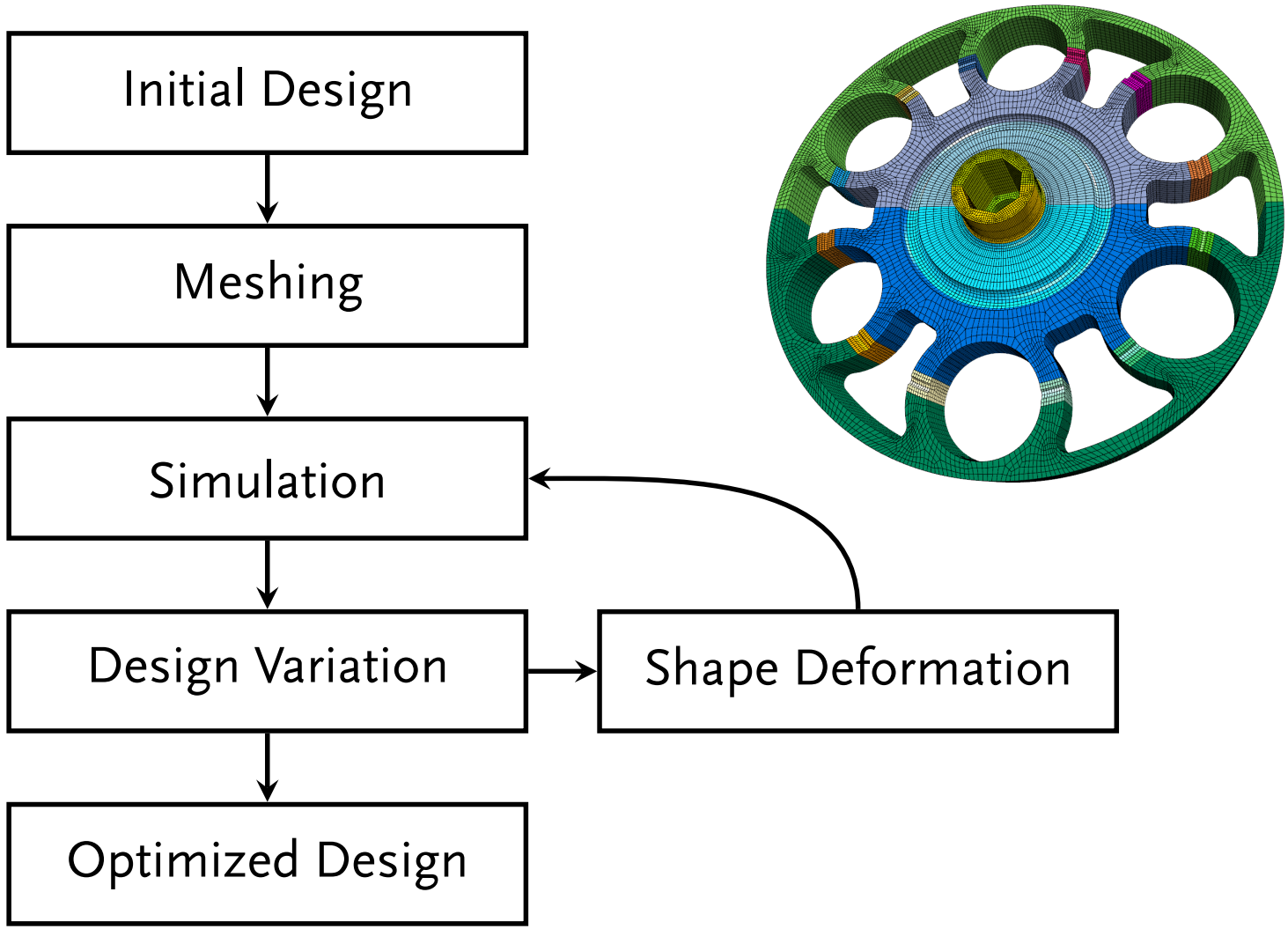




Generalized Design Optimization Scenario



Generalized Design Optimization Scenario



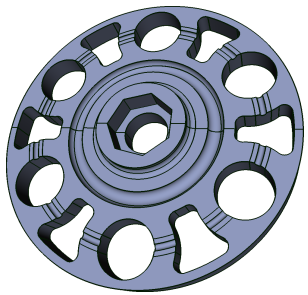
Unified Framework

Procedure:

1. Update CAD model
2. Surface deformation:
   Compute surface points matching updated CAD model
3. Volume deformation:
   Compute volume points matching updated surface points

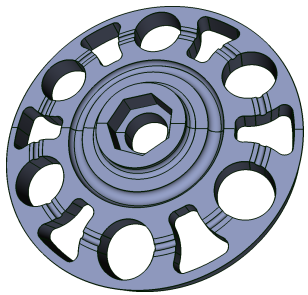
Surface Deformation

CAD

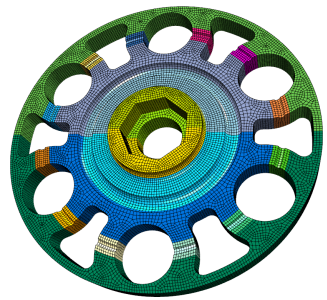


Surface Deformation

CAD

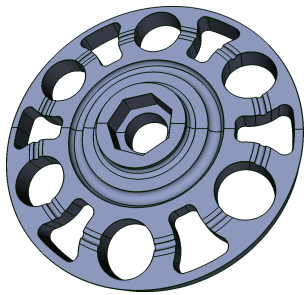


Mesh

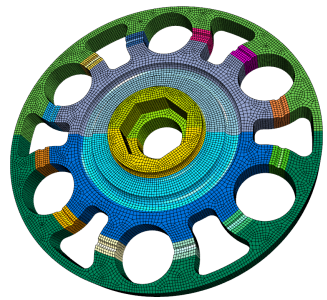


Surface Deformation

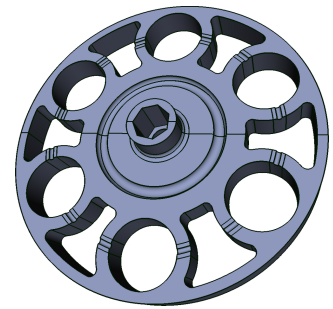
CAD



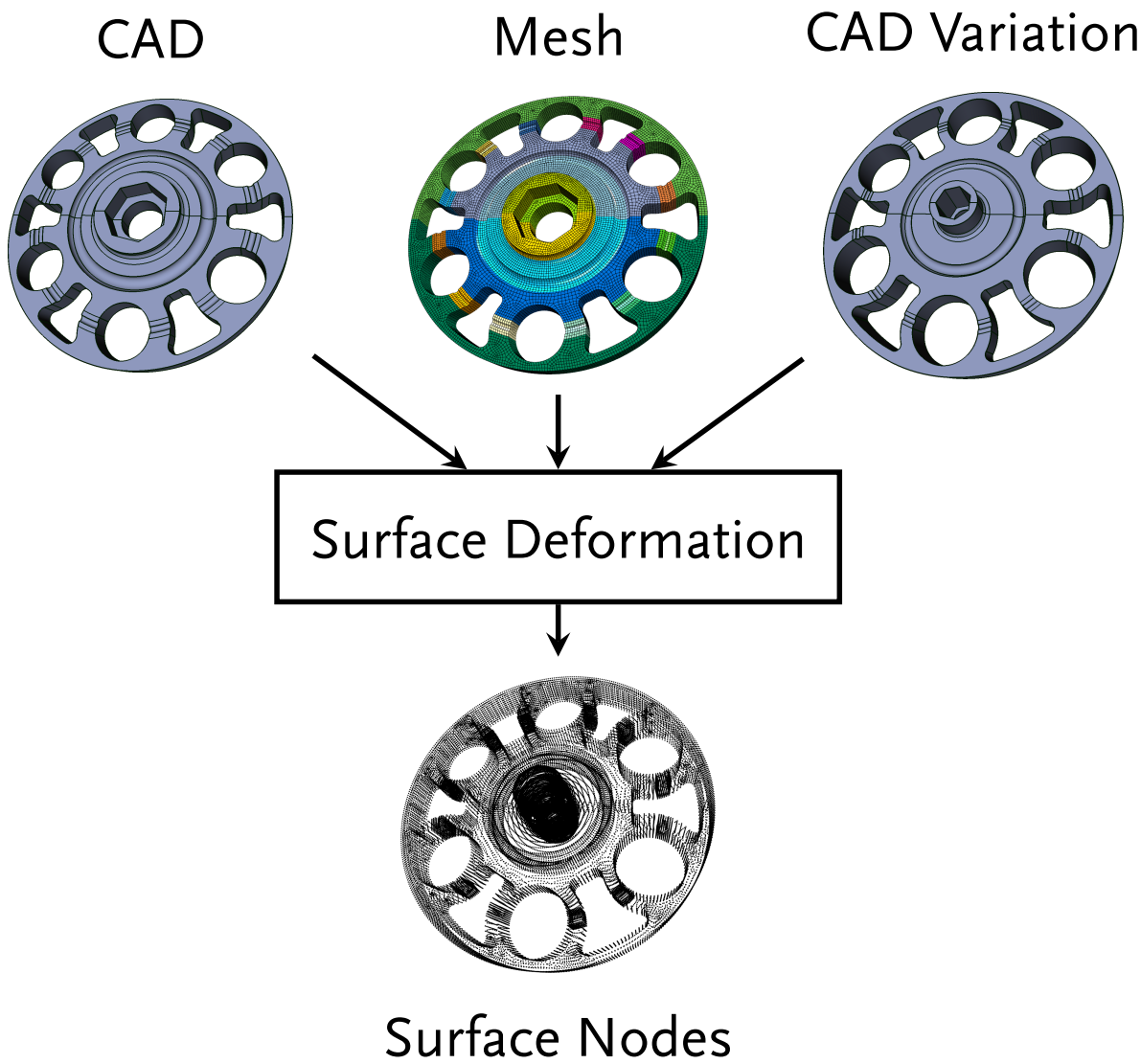
Mesh



CAD Variation

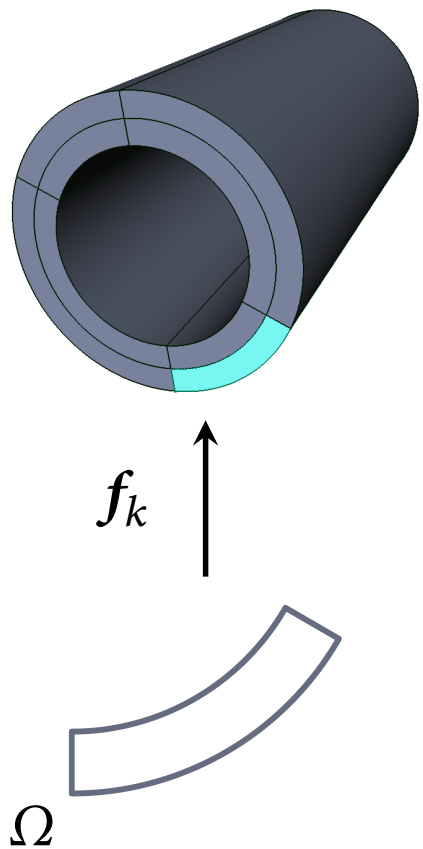


Surface Deformation



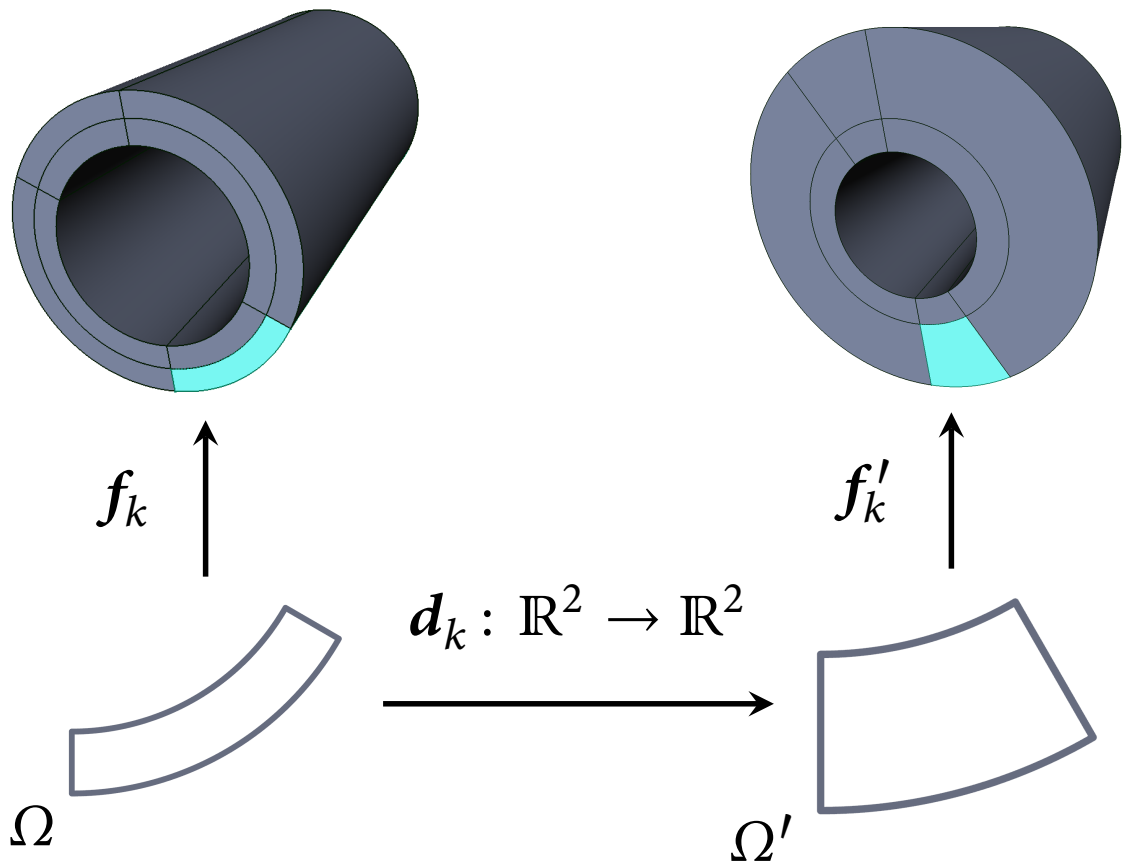
Surface Deformation

1. Determine parametric coordinates for each surface node

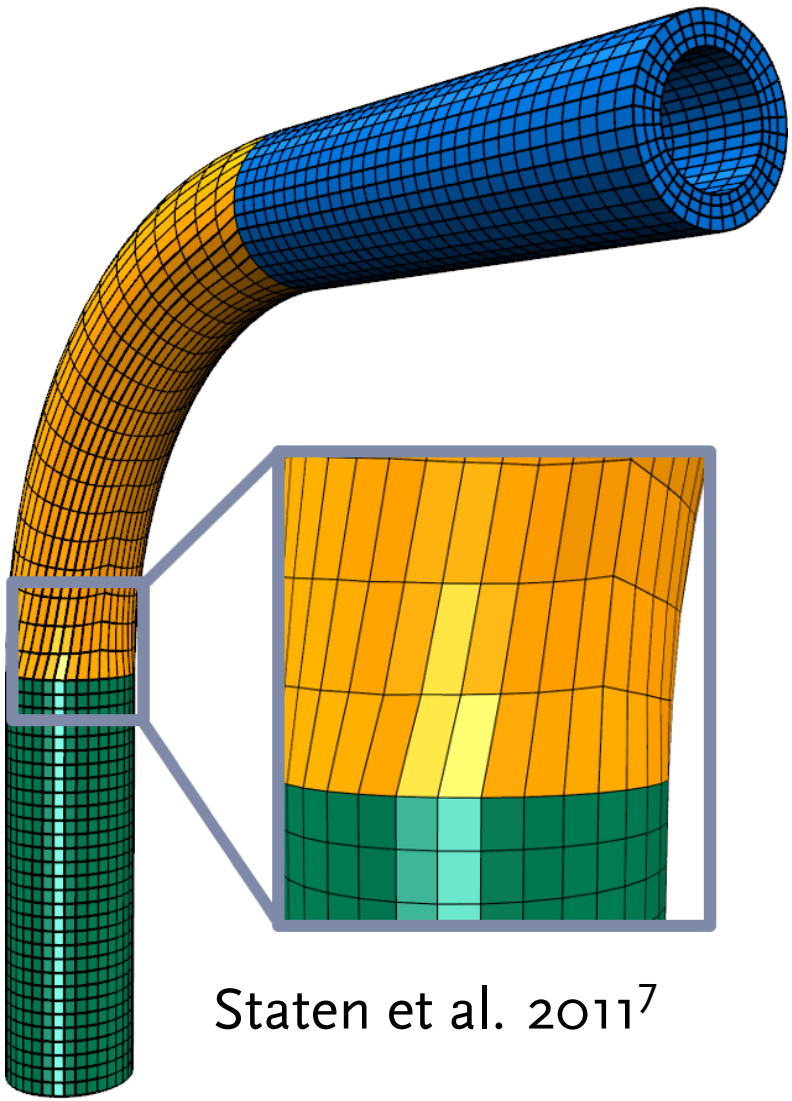


Surface Deformation

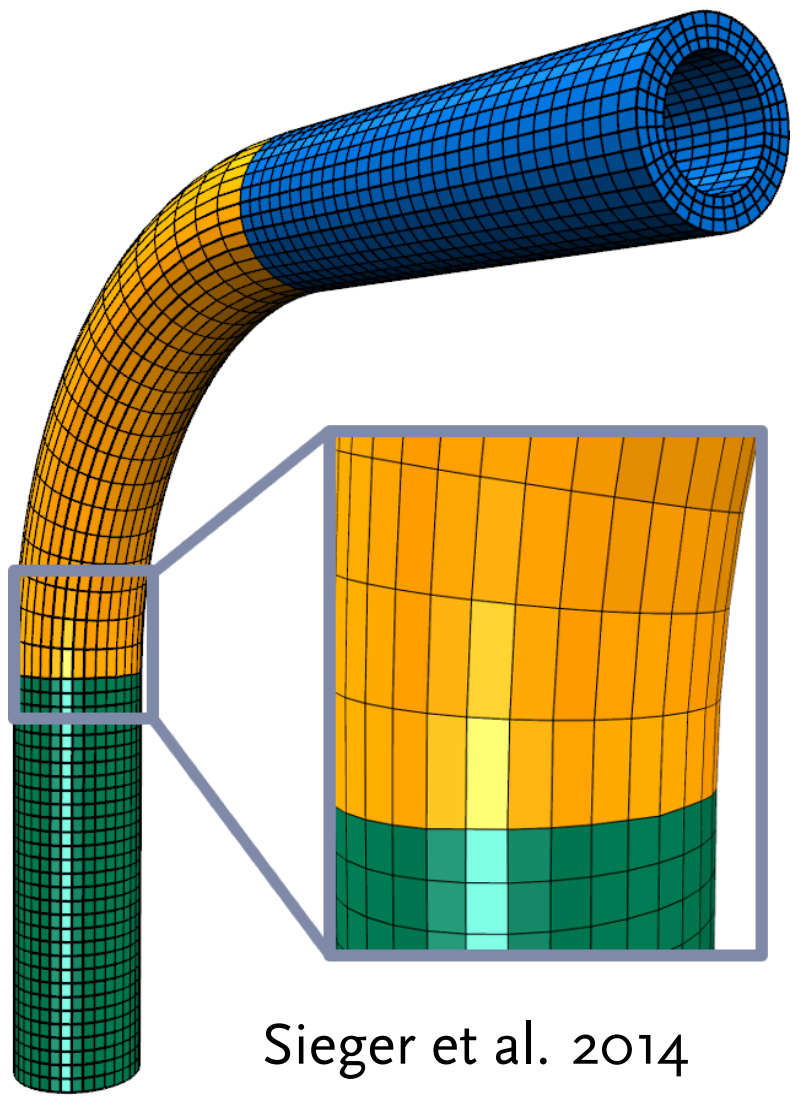
1. Determine parametric coordinates for each surface node
2. Construct 2D RBF deformation to warp coordinates



Surface Deformation



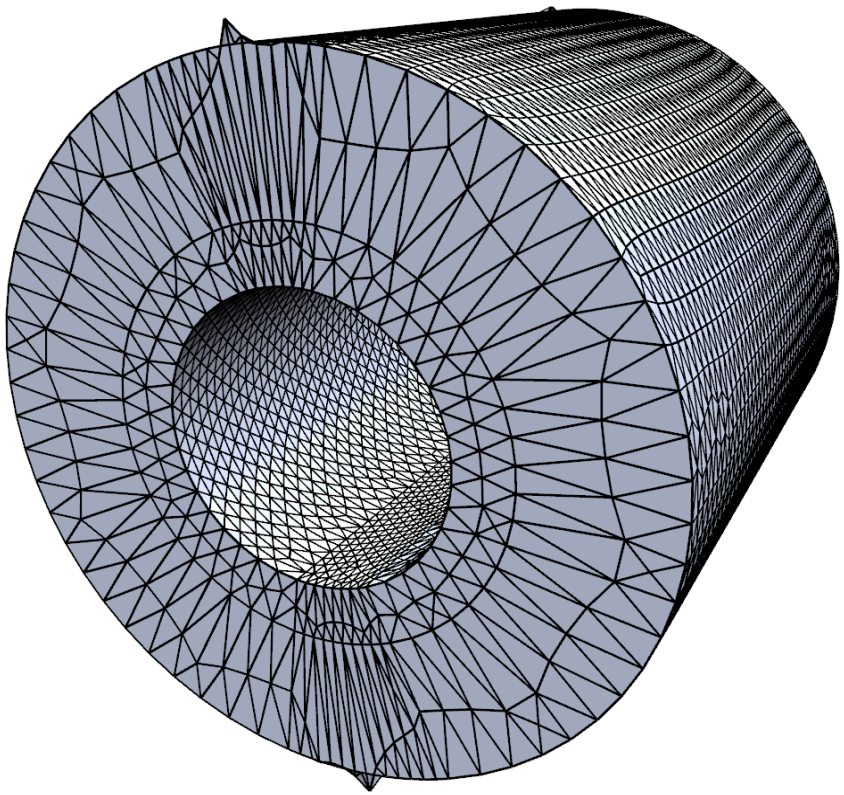
Staten et al. 2011[7]



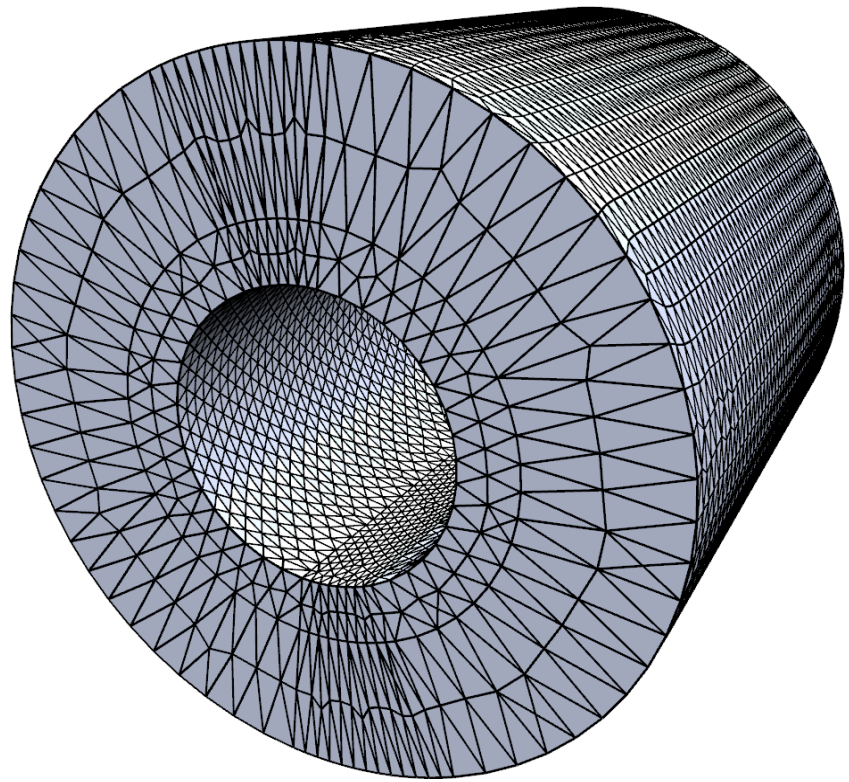
Sieger et al. 2014

7. Staten et al., *A Comparison of Mesh Morphing Methods for 3D Shape Optimization*, IMR 2011

Surface Deformation

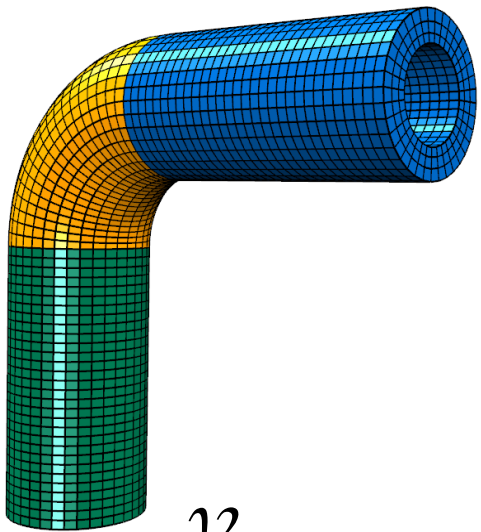


Staten et al. 2011



Sieger et al. 2014

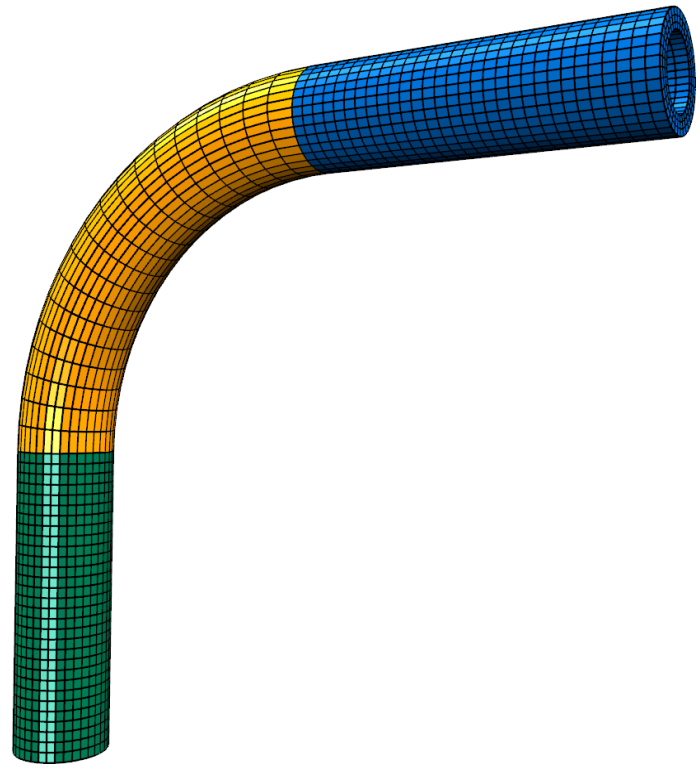
Volume Deformation



$\mathcal{V}$

$$\boldsymbol{d}\colon \mathbb{R}^3 \to \mathbb{R}^3$$

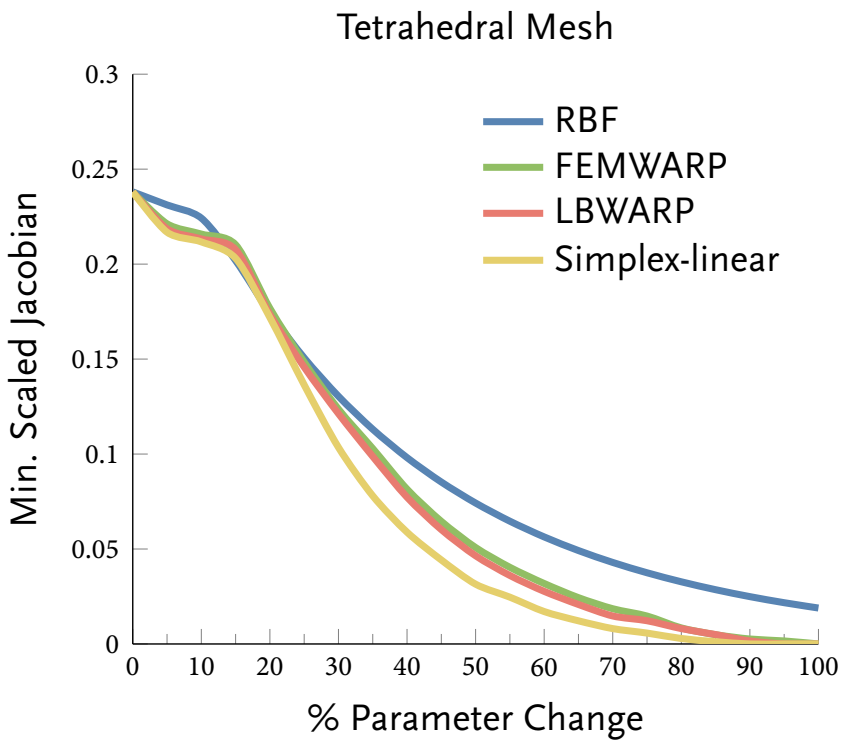
$$\boldsymbol{x}' = \boldsymbol{x} + \boldsymbol{d}(\boldsymbol{x})$$

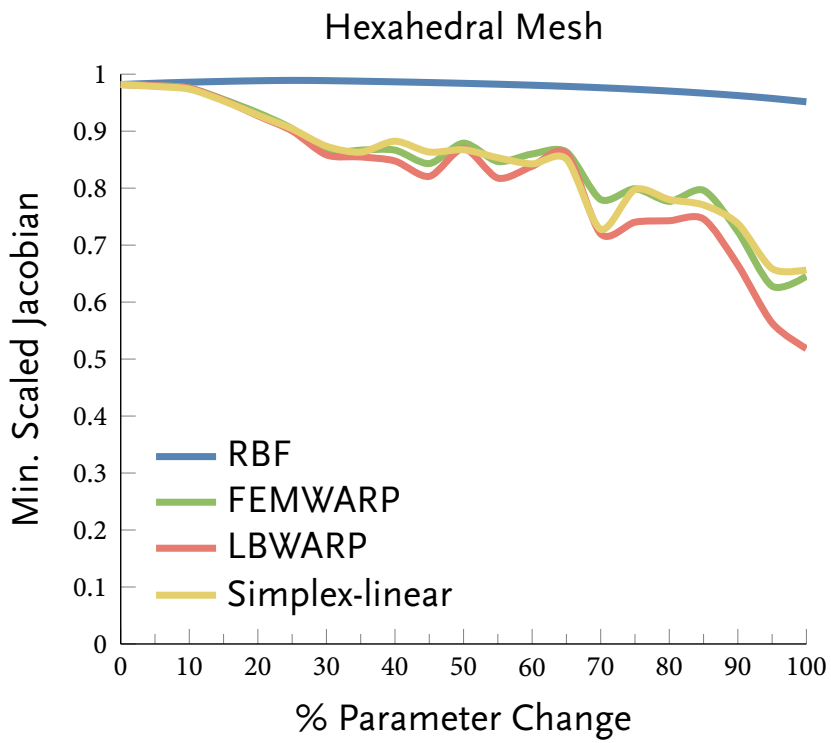


$\mathcal{V}'$

Volume Deformation Quality

- Comparison with best techniques from Staten et al. 2011
- Min. scaled Jacobian vs. parameter change in the CAD model
- Tetrahedral and hexahedral volume meshes





Key Questions and Answers

1. What is a *good* deformation technique for design optimization?

   Kernel-based space deformations provide the quality, flexibility, and robustness required for design optimization.

2. How to *apply* and *improve* existing techniques?

   By *exploiting the flexibility of RBFs*, we can implement fully automatic design optimization loops based on CAD prototypes.

Key Questions and Answers

1. What is a *good* deformation technique for design optimization?

   Kernel-based space deformations provide the quality, flexibility, and robustness required for design optimization.

2. How to *apply* and *improve* existing techniques?

   By *exploiting the flexibility of RBFs*, we can implement fully automatic design optimization loops based on CAD prototypes.

3. How to incorporate *geometric constraints* into the deformation?

Constrained Deformation

Constrained Space Deformation for Design Optimization[7]

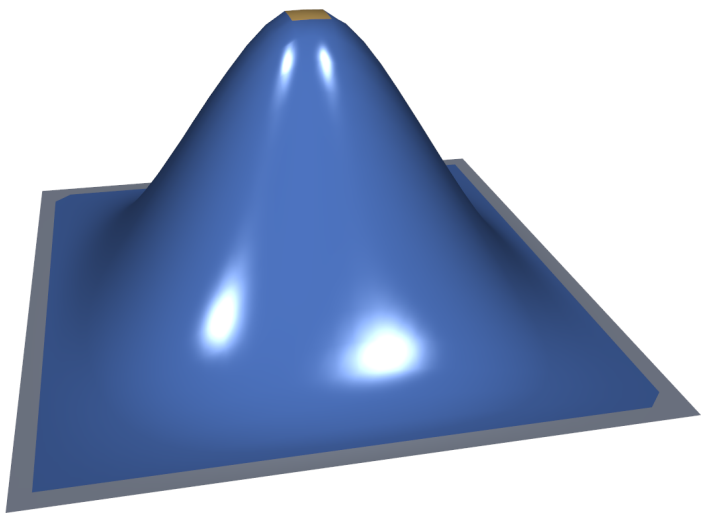
Combines

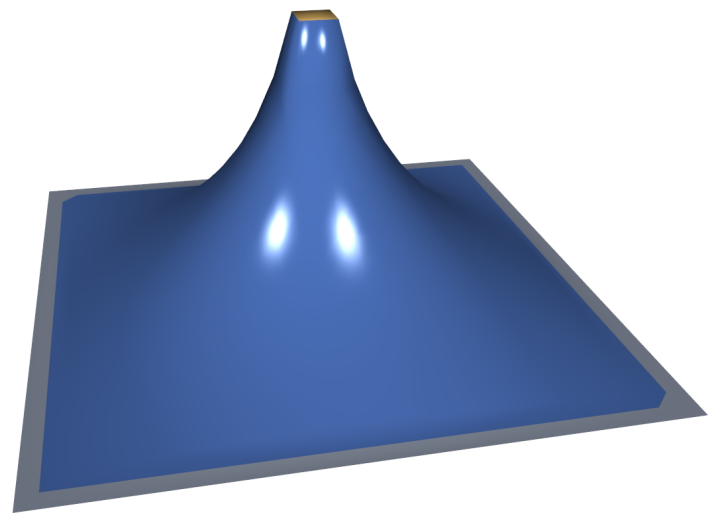
- Modeling flexibility of surface-based methods
- Representation-independence of space deformations
- Quality of RBFs
- Improved scalability
- Geometric constraints

7. Sieger et al., *Constrained Space Deformation for Design Optimization*, Computer-Aided Design 2016

Surface Deformation Revisited

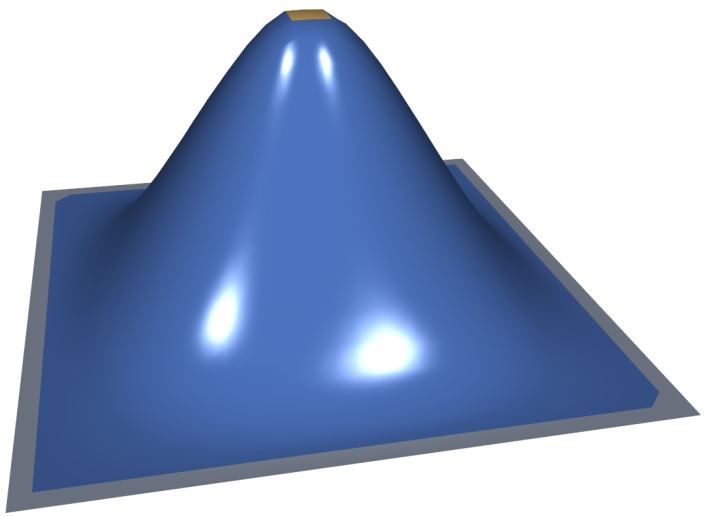
- Surface-based methods allow for flexible modeling of material behavior by explicitly choosing the deformation energy to be minimized

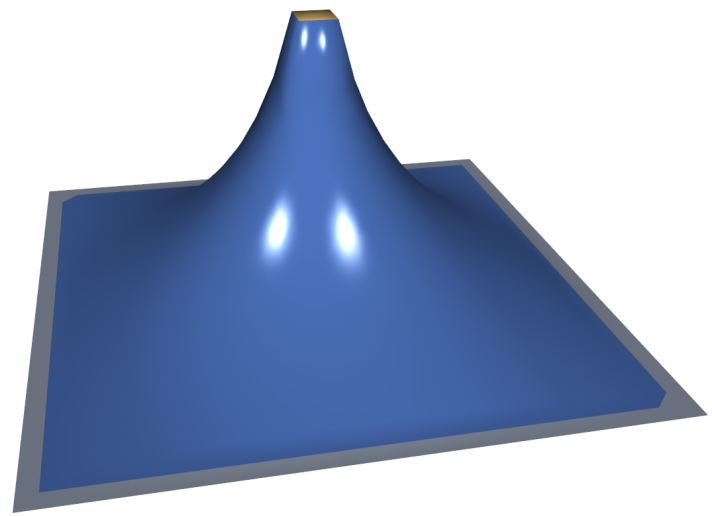




Surface Deformation Revisited

- Surface-based methods allow for flexible modeling of material behavior by explicitly choosing the deformation energy to be minimized
- Goal: Achieve the same flexibility using space deformations





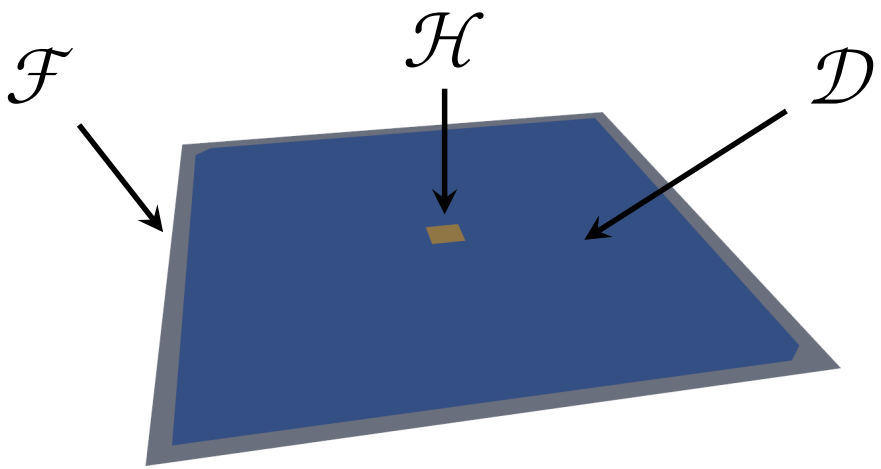
Thin Shell Deformation

Measure stretching and bending by 1st and 2nd order partial derivatives of the displacement function $\boldsymbol{d} : S \to \mathbb{R}^3$

$$E_{\text{stretch}}[\boldsymbol{d}] = \int_{\mathcal{D}} \|\nabla \boldsymbol{d}(\boldsymbol{x})\|^2 \, \mathrm{d}\boldsymbol{x}$$

$$E_{\text{bend}}[\boldsymbol{d}] = \int_{\mathcal{D}} \|\Delta \boldsymbol{d}(\boldsymbol{x})\|^2 \, \mathrm{d}\boldsymbol{x}$$

$$E_{\text{fix}}[\boldsymbol{d}] = \int_{\mathcal{H} \cup \mathcal{F}} \left\|\boldsymbol{d}(\boldsymbol{x}) - \bar{\boldsymbol{d}}(\boldsymbol{x})\right\|^2 \, \mathrm{d}\boldsymbol{x}$$



Surface-Based Discretization

- Discretize on the mesh using standard differential operators [8]

$$E_{\text{stretch}}[\boldsymbol{d}] = \sum_{t \in \mathcal{D}} A_t \, \|\nabla \boldsymbol{\delta}_t\|^2$$

$$E_{\text{bend}}[\boldsymbol{d}] = \sum_{\boldsymbol{x}_i \in \mathcal{D}} A_i \, \|\Delta \boldsymbol{\delta}_i\|^2$$

$$E_{\text{fix}}[\boldsymbol{d}] = \sum_{\boldsymbol{x}_i \in \mathcal{H} \cup \mathcal{F}} A_i \, \left\|\boldsymbol{\delta}_i - \bar{\boldsymbol{\delta}}_i\right\|^2$$

➔ Solve a linear system:

$$\left[\gamma_s \boldsymbol{G}^{\mathrm{T}}\boldsymbol{G} + \gamma_b \boldsymbol{L}^{\mathrm{T}}\boldsymbol{L} + \gamma_f \boldsymbol{F}^{\mathrm{T}}\boldsymbol{F}\right]\boldsymbol{D} \;=\; \gamma_f \boldsymbol{F}^{\mathrm{T}}\boldsymbol{F}\bar{\boldsymbol{D}}$$

8. Meyer et al., *Discrete Differential-Geometry Operators for Triangulated 2-Manifolds*, Visualization and Mathematics, 2003

Space Deformation Methods

- Construct a space deformation function $\boldsymbol{d}: \Omega \subset \mathbb{R}^3 \rightarrow \mathbb{R}^3$
  - Use meshless approximation methods
  - Represent $\boldsymbol{d}$ by basis functions φ_j located at centers $\boldsymbol{c}_j$:

$$\boldsymbol{d}(\boldsymbol{x}) = \sum_{j=1}^{k} \boldsymbol{w}_j \varphi_j(\boldsymbol{x})$$

Space Deformation Methods

- Construct a space deformation function $\boldsymbol{d}: \Omega \subset \mathbb{R}^3 \rightarrow \mathbb{R}^3$
  - Use meshless approximation methods
  - Represent $\boldsymbol{d}$ by basis functions φ_j located at centers $\boldsymbol{c}_j$:

$$\boldsymbol{d}(\boldsymbol{x}) = \sum_{j=1}^{k} \boldsymbol{w}_j \varphi_j(\boldsymbol{x})$$

- Questions:
  - Which basis functions to choose?
  - Where to place basis functions?

Space Deformation Methods

- Construct a space deformation function $\boldsymbol{d}: \Omega \subset \mathbb{R}^3 \rightarrow \mathbb{R}^3$
  - Use meshless approximation methods
  - Represent $\boldsymbol{d}$ by basis functions φ_j located at centers $\boldsymbol{c}_j$:

$$\boldsymbol{d}(\boldsymbol{x}) = \sum_{j=1}^{k} \boldsymbol{w}_j \varphi_j(\boldsymbol{x})$$

- Questions:
  - Which basis functions to choose?
  - Where to place basis functions?

Space Deformation Methods

- Construct a space deformation function $\boldsymbol{d}: \Omega \subset \mathbb{R}^3 \rightarrow \mathbb{R}^3$
  - Use meshless approximation methods
  - Represent $\boldsymbol{d}$ by basis functions φ_j located at centers $\boldsymbol{c}_j$:

$$\boldsymbol{d}(\boldsymbol{x}) = \sum_{j=1}^{k} \boldsymbol{w}_j \varphi_j(\boldsymbol{x})$$

- Questions:
  - Which basis functions to choose?
  - Where to place basis functions?

Surface Sampling

Goal: Generate uniformly distributed points $\boldsymbol{c}_j$ on the surface

Surface Sampling

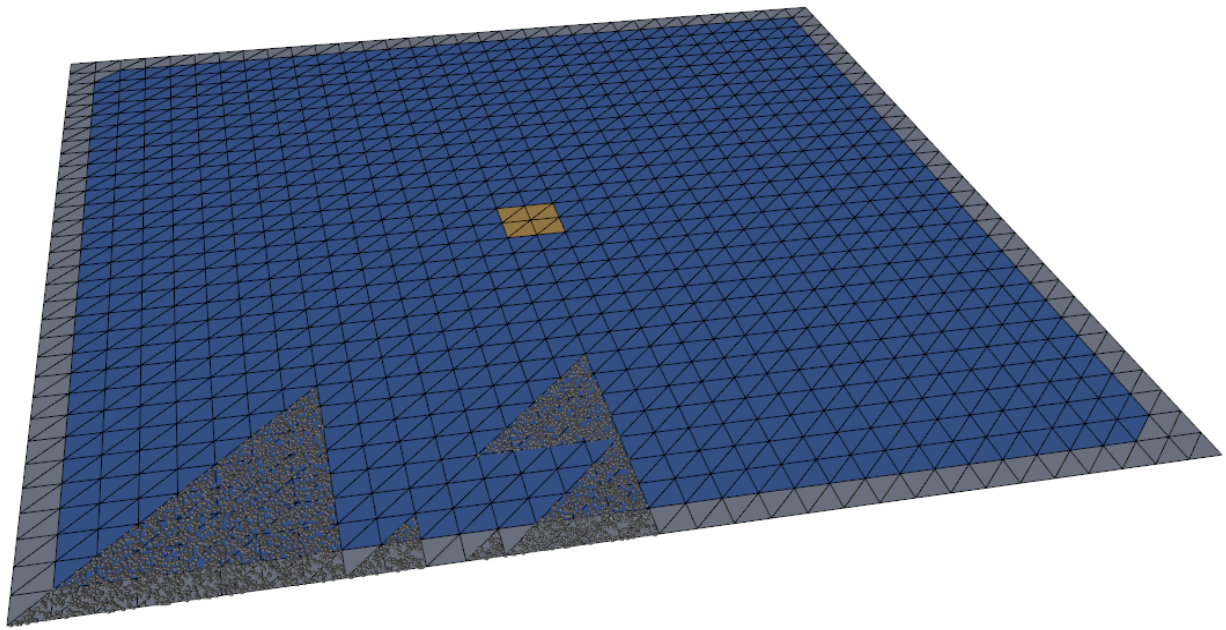
Goal: Generate uniformly distributed points $\boldsymbol{c}_j$ on the surface

1. Dense random sampling of each mesh face

Surface Sampling

Goal: Generate uniformly distributed points $\boldsymbol{c}_j$ on the surface

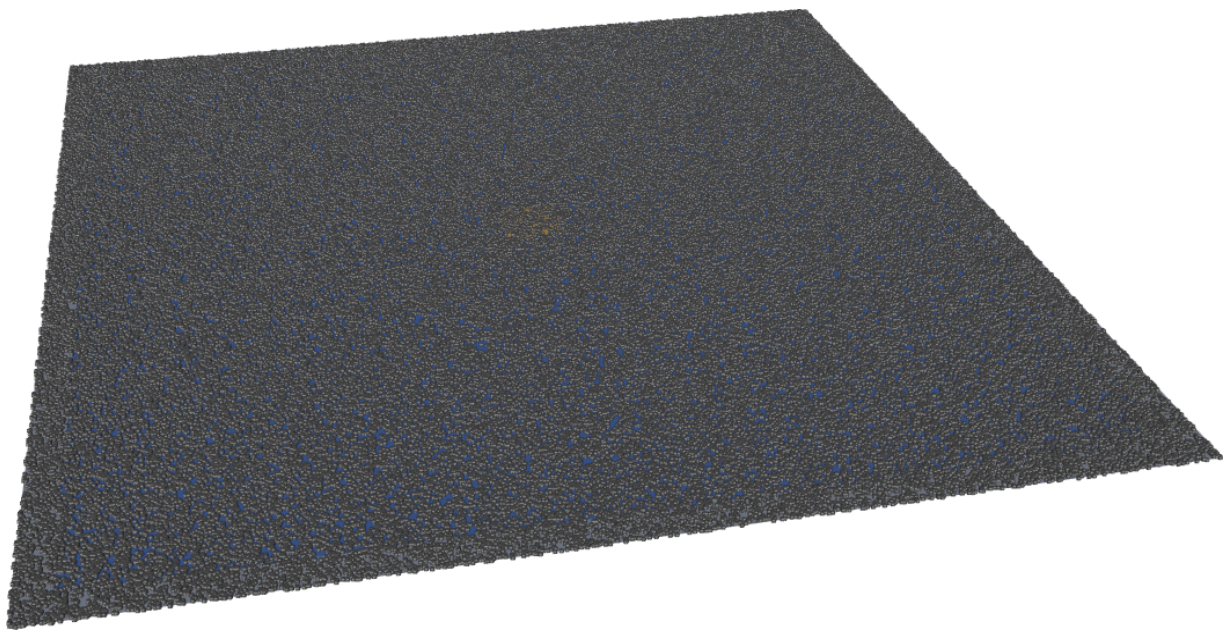
1. Dense random sampling of each mesh face



Surface Sampling

Goal: Generate uniformly distributed points $\boldsymbol{c}_j$ on the surface

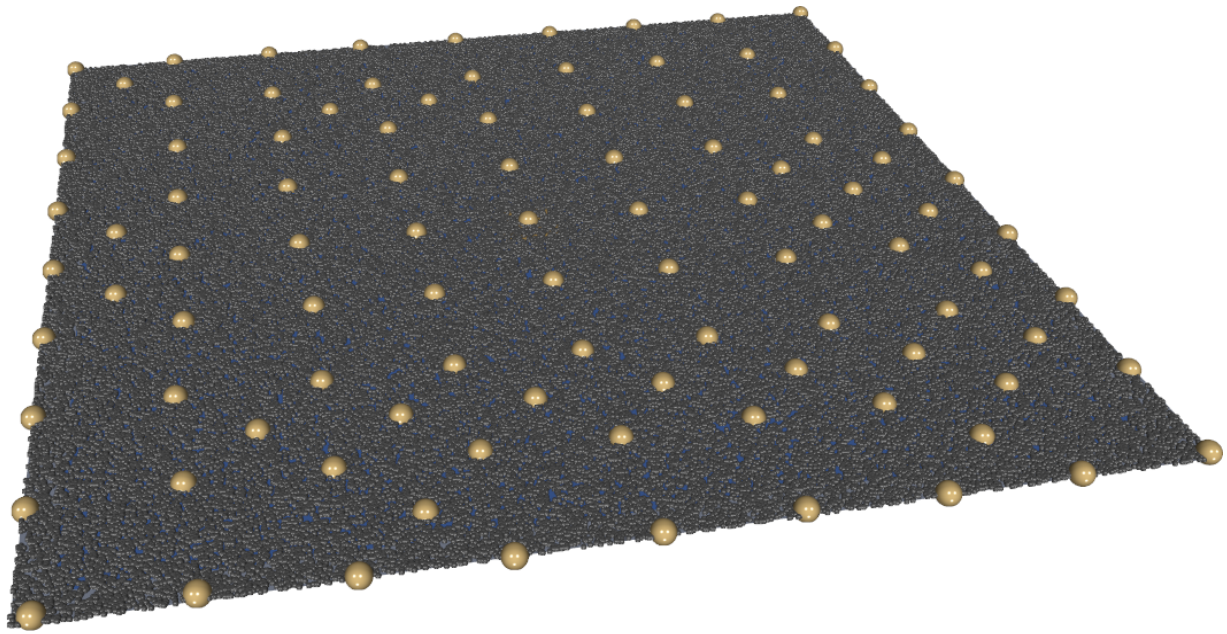
1. Dense random sampling of each mesh face



Surface Sampling

Goal: Generate uniformly distributed points c_j on the surface

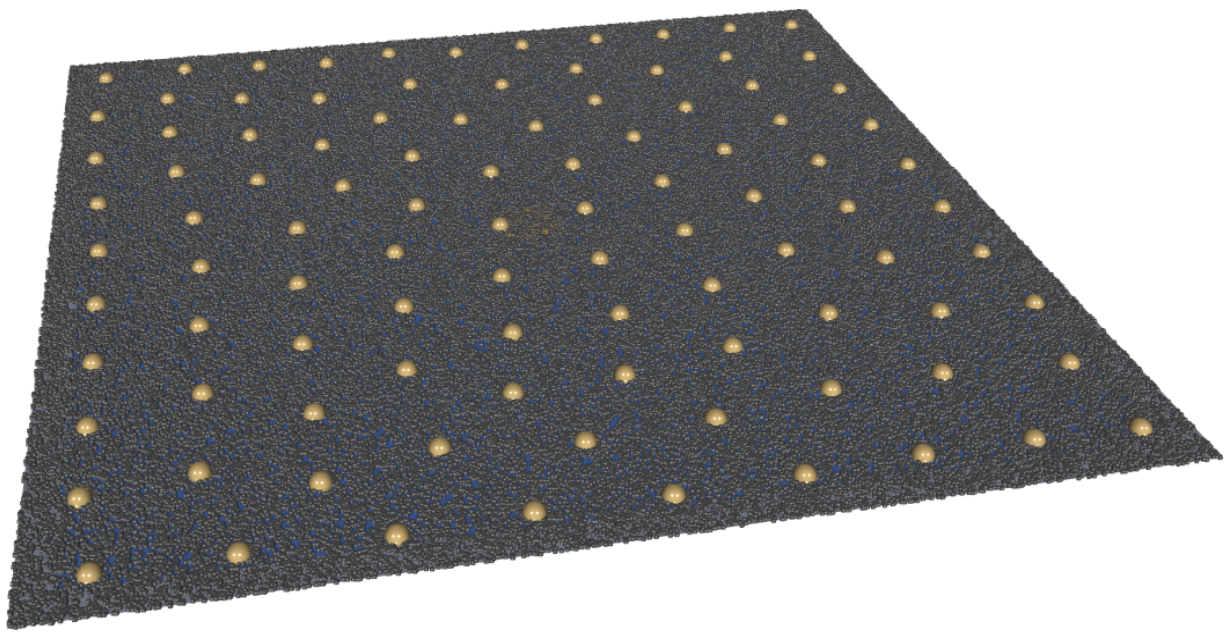
1. Dense random sampling of each mesh face
2. Farthest point selection



Surface Sampling

Goal: Generate uniformly distributed points $\boldsymbol{c}_j$ on the surface

1. Dense random sampling of each mesh face
2. Farthest point selection
3. Lloyd relaxation (k-means clustering)



Subspace Surface Deformation

- Which basis functions φ_j to choose?

Subspace Surface Deformation

- Which basis functions φ_j to choose?
- Goal: Achieve same modeling flexibility as surface deformation

Subspace Surface Deformation

- Which basis functions φ_j to choose?
- Goal: Achieve same modeling flexibility as surface deformation
- Ground truth: Combine surface energy with space deformation

Subspace Surface Deformation

- Which basis functions φ_j to choose?
- Goal: Achieve same modeling flexibility as surface deformation
- Ground truth: Combine surface energy with space deformation
- Express $\boldsymbol{D}$ through coefficients $\boldsymbol{W}$ and subspace matrix $\boldsymbol{\Phi}$

$$\boldsymbol{D} = \boldsymbol{\Phi}\boldsymbol{W} \quad \text{with} \quad \boldsymbol{\Phi}_{ij} = \varphi_j(\boldsymbol{x}_i)$$

Subspace Surface Deformation

- Which basis functions φ_j to choose?
- Goal: Achieve same modeling flexibility as surface deformation
- Ground truth: Combine surface energy with space deformation
- Express $\boldsymbol{D}$ through coefficients $\boldsymbol{W}$ and subspace matrix $\boldsymbol{\Phi}$

$$\boldsymbol{D} = \boldsymbol{\Phi}\boldsymbol{W} \quad \text{with} \quad \boldsymbol{\Phi}_{ij} = \varphi_j(\boldsymbol{x}_i)$$

- Leads to a modified linear system:

$$\boldsymbol{\Phi}^{\mathrm{T}} \left[\gamma_s \boldsymbol{G}^{\mathrm{T}}\boldsymbol{G} + \gamma_b \boldsymbol{L}^{\mathrm{T}}\boldsymbol{L} + \gamma_f \boldsymbol{F}^{\mathrm{T}}\boldsymbol{F} \right] \boldsymbol{\Phi}\,\boldsymbol{W} \;=\; \boldsymbol{\Phi}^{\mathrm{T}} \left[\gamma_f \boldsymbol{F}^{\mathrm{T}}\boldsymbol{F}\bar{\boldsymbol{D}} \right]$$

Subspace Surface Deformation

- Global triharmonic RBFs

$$\varphi_j(\boldsymbol{x}) \;=\; \left\| \boldsymbol{x} - \boldsymbol{c}_j \right\|^3$$

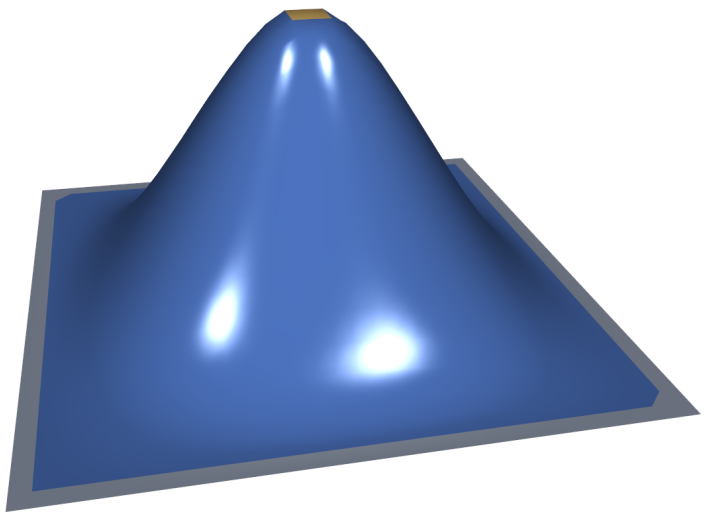
- Energy minimization built-in

Subspace Surface Deformation

- Global triharmonic RBFs

$$\varphi_j(\boldsymbol{x}) \;=\; \left\|\boldsymbol{x} - \boldsymbol{c}_j\right\|^3$$

- Energy minimization built-in
- Good results for bending

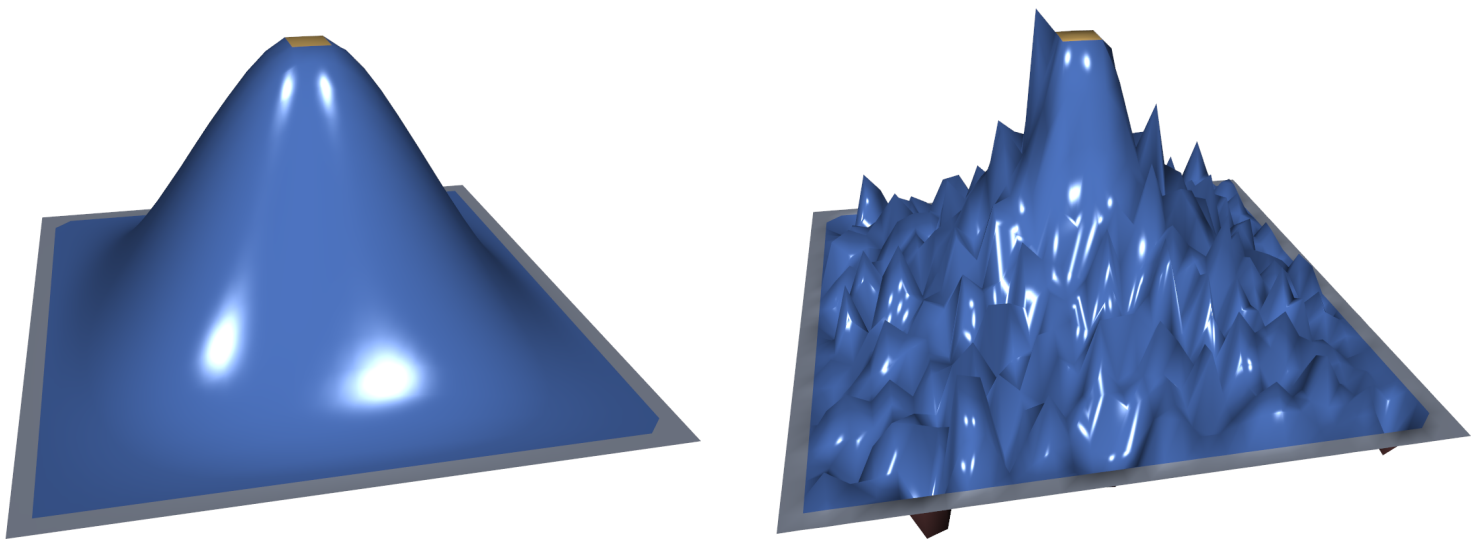


Subspace Surface Deformation

- Global triharmonic RBFs

$$\varphi_j(\boldsymbol{x}) \;=\; \left\|\boldsymbol{x}-\boldsymbol{c}_j\right\|^3$$

- Energy minimization built-in
- Good results for bending, bad results for stretching

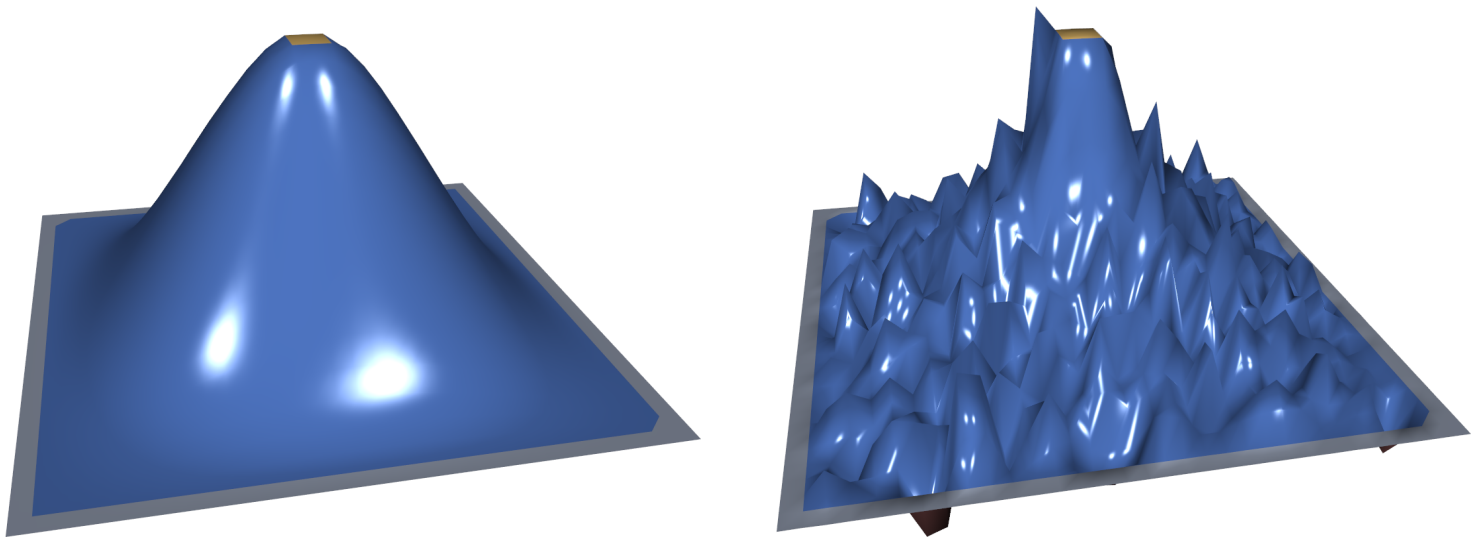


Subspace Surface Deformation

- Global triharmonic RBFs

$$\varphi_j(\boldsymbol{x}) \;=\; \left\|\boldsymbol{x} - \boldsymbol{c}_j\right\|^3$$

- Energy minimization built-in
- Good results for bending, bad results for stretching
- Global support → dense linear systems



Subspace Surface Deformation

- Compactly supported RBFs

$$\varphi_j(\boldsymbol{x}) \;=\; \varphi\left(\left\|\boldsymbol{x}-\boldsymbol{c}_j\right\|\right) \;=\; \varphi(r) \;=\; \begin{cases} (1-r)^4(4r+1)\,, & r<\sigma\,, \\ 0\,, & \text{otherwise}\,. \end{cases}$$

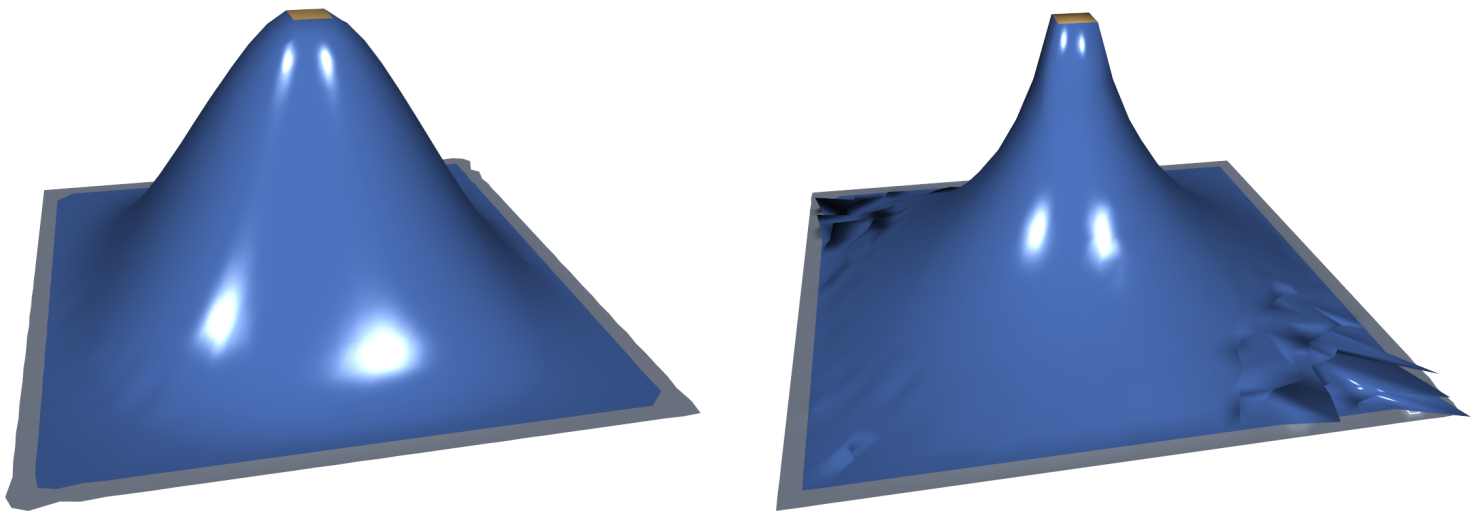
- No built-in energy minimization

Subspace Surface Deformation

- Compactly supported RBFs

$$\varphi_j(\boldsymbol{x}) \;=\; \varphi\left(\left\|\boldsymbol{x}-\boldsymbol{c}_j\right\|\right) \;=\; \varphi(r) \;=\; \begin{cases} (1-r)^4(4r+1)\,, & r<\sigma\,, \\ 0\,, & \text{otherwise}\,. \end{cases}$$

- No built-in energy minimization
- *Small* support → *sparse* linear system, *bad* results

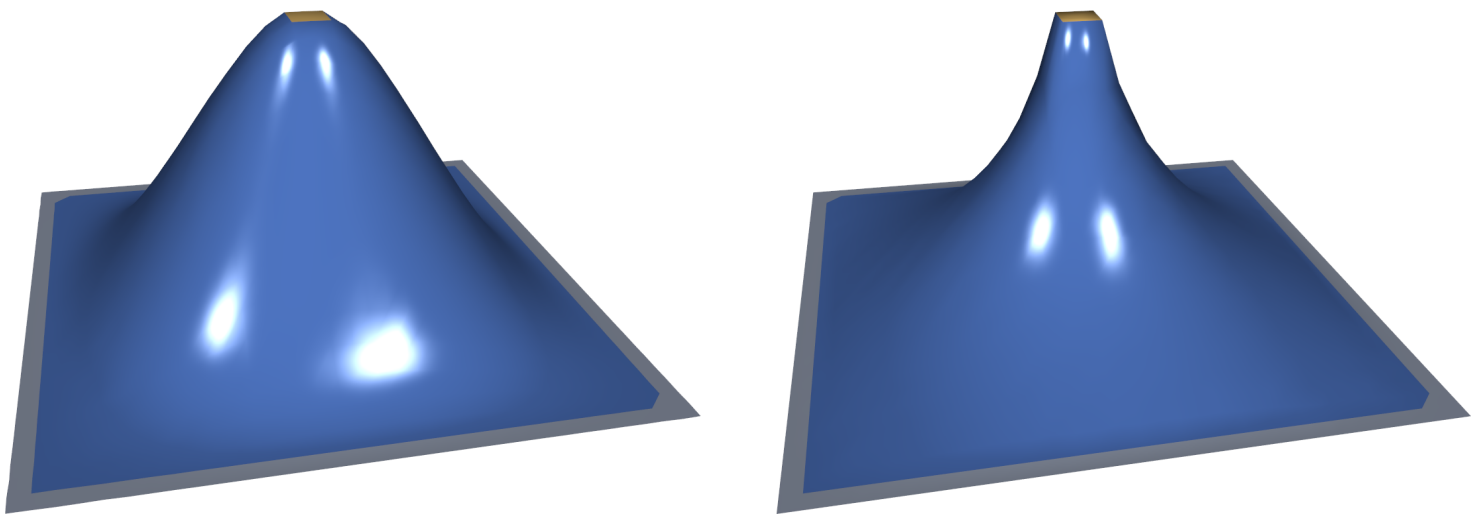


Subspace Surface Deformation

- Compactly supported RBFs

$$\varphi_j(\boldsymbol{x}) \;=\; \varphi\left(\left\|\boldsymbol{x}-\boldsymbol{c}_j\right\|\right) \;=\; \varphi(r) \;=\; \begin{cases} (1-r)^4(4r+1)\,, & r<\sigma\,, \\ 0\,, & \text{otherwise}\,. \end{cases}$$

- No built-in energy minimization
- *Large* support → *dense* linear system, *good* results



Subspace Surface Deformation

- Moving Least Squares (MLS) basis functions[9]

$$\varphi_j(\boldsymbol{x}) \;=\; \boldsymbol{p}(\boldsymbol{x})^{\mathrm{T}} \boldsymbol{M}^{-1}(\boldsymbol{x}) \boldsymbol{p}(\boldsymbol{c}_j) w(\boldsymbol{x} - \boldsymbol{c}_j)$$

9. Fries et al., *Classification and Overview of Meshfree Methods*, Technical Report, TU Braunschweig, 2003

Subspace Surface Deformation

- Moving Least Squares (MLS) basis functions[9]

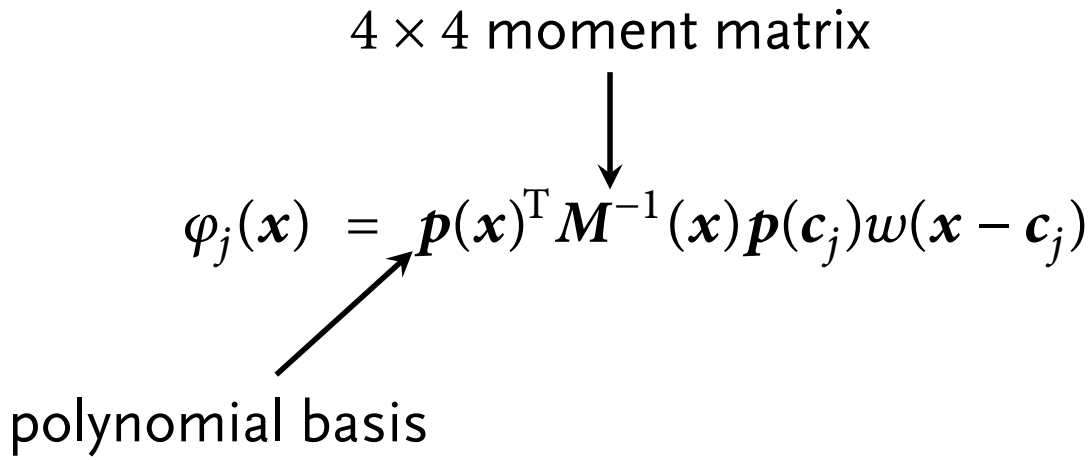
$$\varphi_j(\boldsymbol{x}) = \boldsymbol{p}(\boldsymbol{x})^{\mathrm{T}} \boldsymbol{M}^{-1}(\boldsymbol{x}) \boldsymbol{p}(\boldsymbol{c}_j) w(\boldsymbol{x} - \boldsymbol{c}_j)$$

polynomial basis

9. Fries et al., *Classification and Overview of Meshfree Methods*, Technical Report, TU Braunschweig, 2003

Subspace Surface Deformation

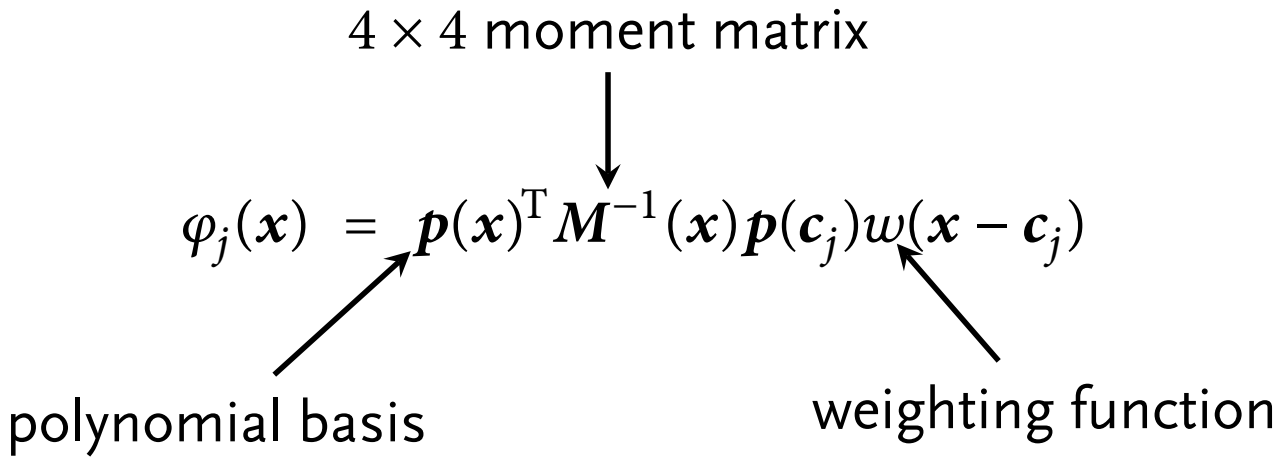
- Moving Least Squares (MLS) basis functions[9]



9. Fries et al., *Classification and Overview of Meshfree Methods*, Technical Report, TU Braunschweig, 2003

Subspace Surface Deformation

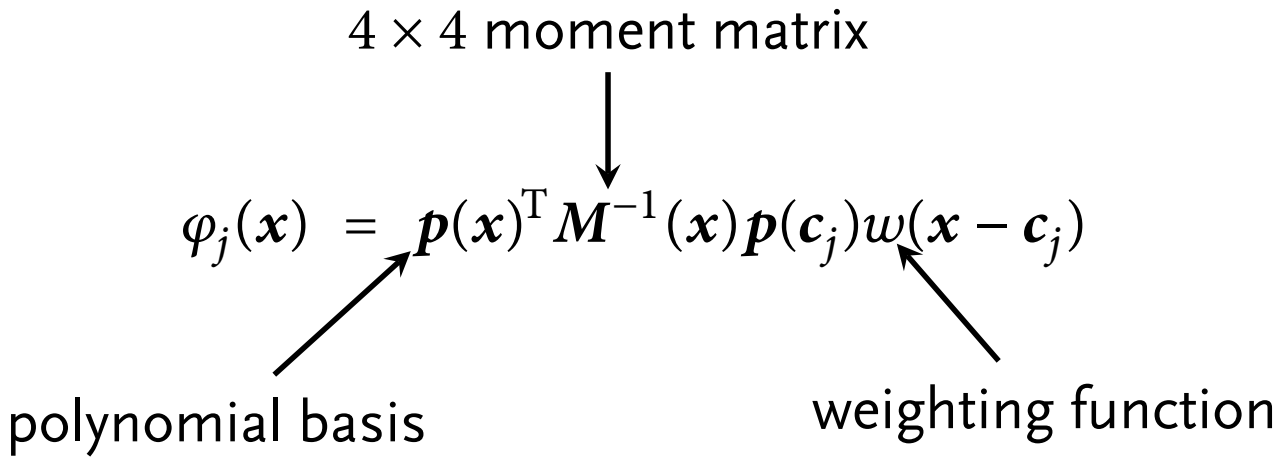
- Moving Least Squares (MLS) basis functions[9]



9. Fries et al., *Classification and Overview of Meshfree Methods*, Technical Report, TU Braunschweig, 2003

Subspace Surface Deformation

- Moving Least Squares (MLS) basis functions[9]

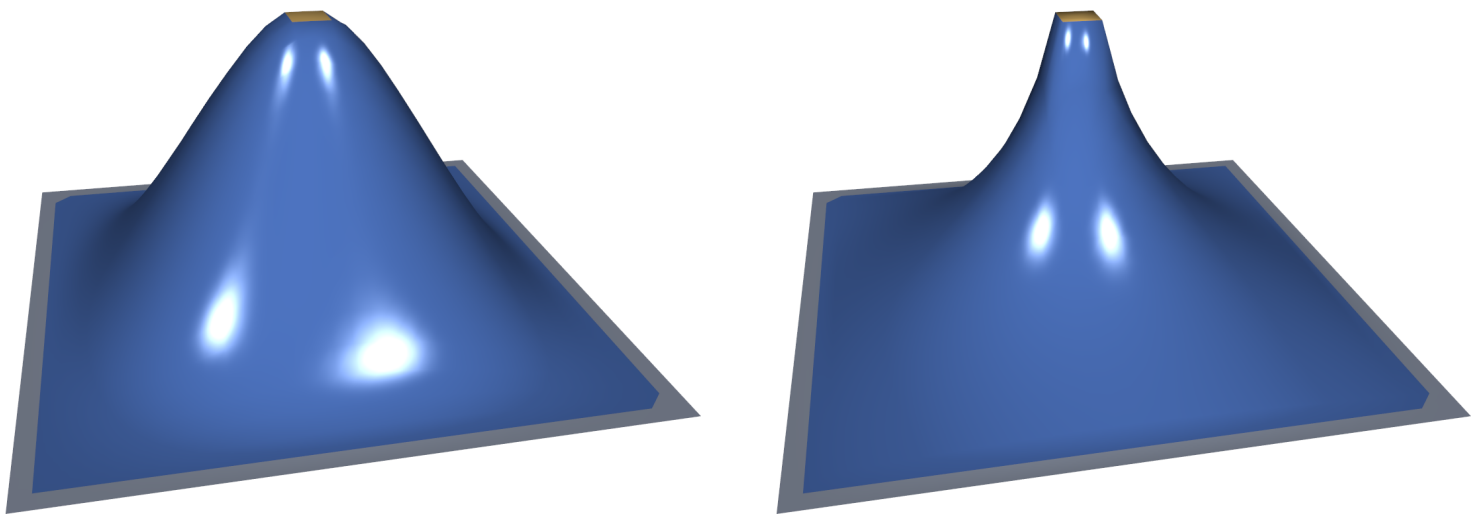


- More complex form, inversion of $\boldsymbol{M}$ required

9. Fries et al., *Classification and Overview of Meshfree Methods*, Technical Report, TU Braunschweig, 2003

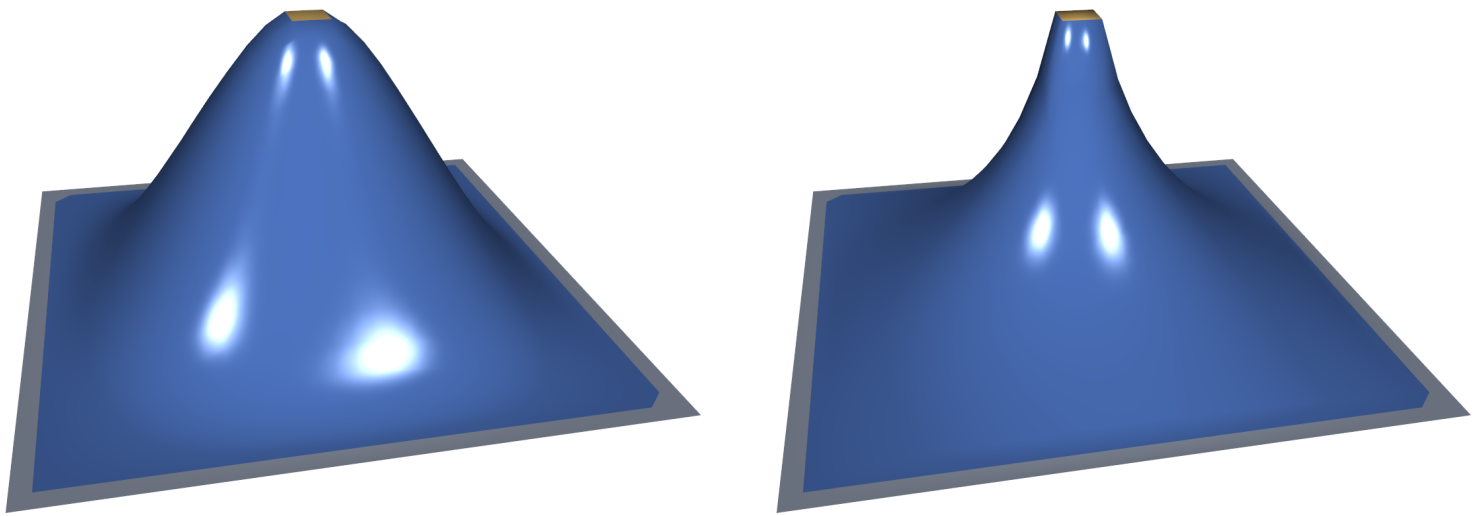
Subspace Surface Deformation

- Moving Least Squares (MLS) basis functions
- \+ *Small* support → *sparse* linear system, *good* results



Subspace Surface Deformation

- Moving Least Squares (MLS) basis functions
- \+ *Small* support → *sparse* linear system, *good* results



Volumetric Space Deformation

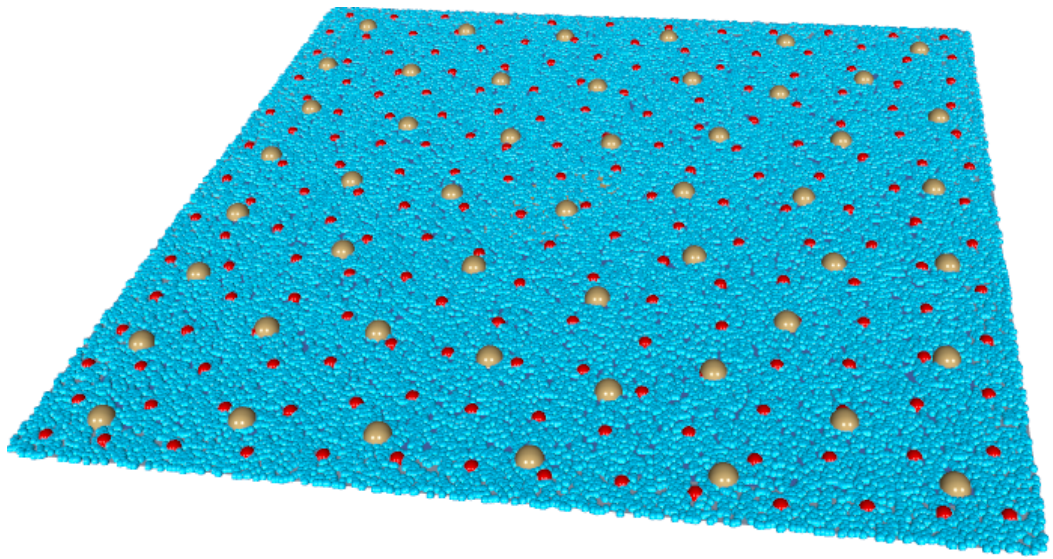
- Goal: Fully space-based discretization of stretching and bending energies using MLS approximation

Volumetric Space Deformation

- Goal: Fully space-based discretization of stretching and bending energies using MLS approximation
- Replace vertex-based integration over the surface $\mathcal{S}$ with purely space-based integration method

Volumetric Space Deformation

- Goal: Fully space-based discretization of stretching and bending energies using MLS approximation
- Replace vertex-based integration over the surface $\mathcal{S}$ with purely space-based integration method
- Use Lloyd-based sampling to determine integration points $\boldsymbol{t}_i$



Volumetric Space Deformation

- Evaluate gradients and Laplacians of φ_j at integration points $\boldsymbol{t}_i$:

$$E_{\text{stretch}} = \sum_{i=1}^{N} V_i \left\| \nabla \boldsymbol{d}(\boldsymbol{t}_i) \right\|^2 = \sum_{i=1}^{N} V_i \left\| \sum_{j=1}^{k} \boldsymbol{w}_j \nabla \varphi_j(\boldsymbol{t}_i) \right\|^2$$

$$E_{\text{bend}} = \sum_{i=1}^{N} V_i \left\| \Delta \boldsymbol{d}(\boldsymbol{t}_i) \right\|^2 = \sum_{i=1}^{N} V_i \left\| \sum_{j=1}^{k} \boldsymbol{w}_j \Delta \varphi_j(\boldsymbol{t}_i) \right\|^2$$

Volumetric Space Deformation

- Evaluate gradients and Laplacians of φ_j at integration points $\boldsymbol{t}_i$:

$$E_{\text{stretch}} = \sum_{i=1}^{N} V_i \left\| \nabla \boldsymbol{d}(\boldsymbol{t}_i) \right\|^2 = \sum_{i=1}^{N} V_i \left\| \sum_{j=1}^{k} \boldsymbol{w}_j \nabla \varphi_j(\boldsymbol{t}_i) \right\|^2$$

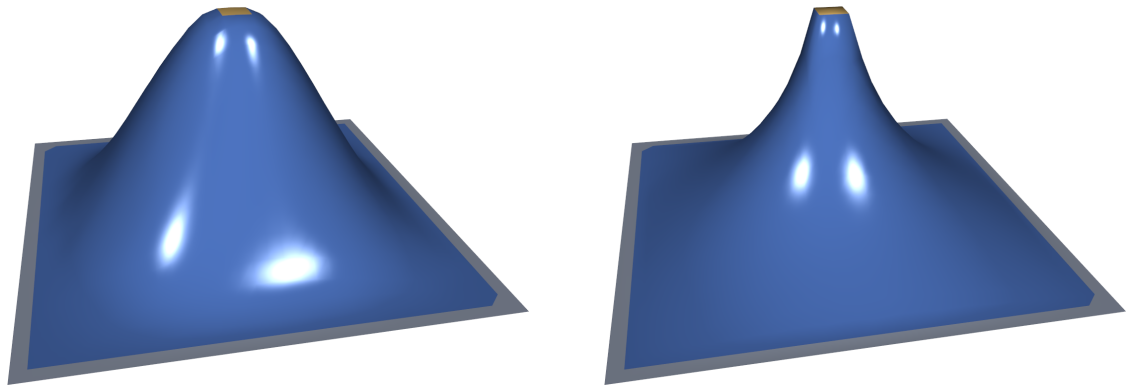
$$E_{\text{bend}} = \sum_{i=1}^{N} V_i \left\| \Delta \boldsymbol{d}(\boldsymbol{t}_i) \right\|^2 = \sum_{i=1}^{N} V_i \left\| \sum_{j=1}^{k} \boldsymbol{w}_j \Delta \varphi_j(\boldsymbol{t}_i) \right\|^2$$

- Leads to a modified linear system

$$\left[\gamma_s \boldsymbol{G}^{\mathrm{T}} \boldsymbol{G} + \gamma_b \boldsymbol{L}^{\mathrm{T}} \boldsymbol{L} + \gamma_f \boldsymbol{\Phi}^{\mathrm{T}} \boldsymbol{F}^{\mathrm{T}} \boldsymbol{F} \boldsymbol{\Phi} \right] \boldsymbol{W} = \gamma_f \boldsymbol{\Phi}^{\mathrm{T}} \boldsymbol{F}^{\mathrm{T}} \boldsymbol{F} \bar{\boldsymbol{D}}$$

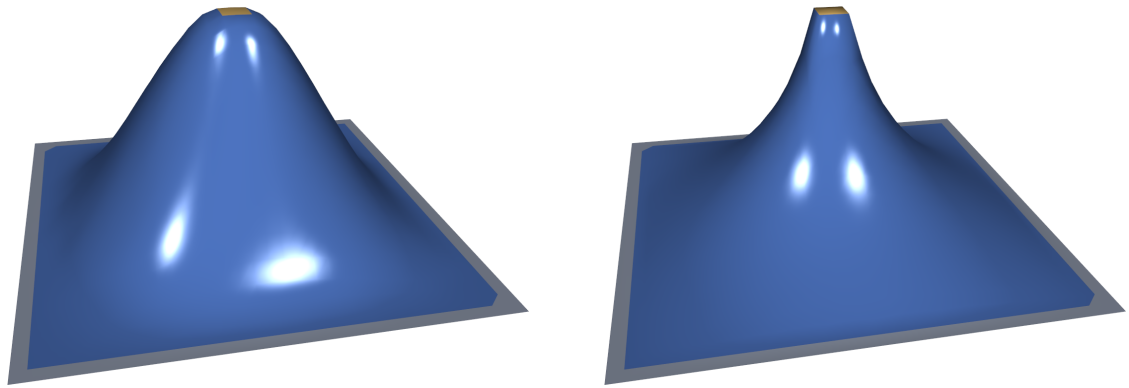
Volumetric Space Deformation

- Space-based discretization

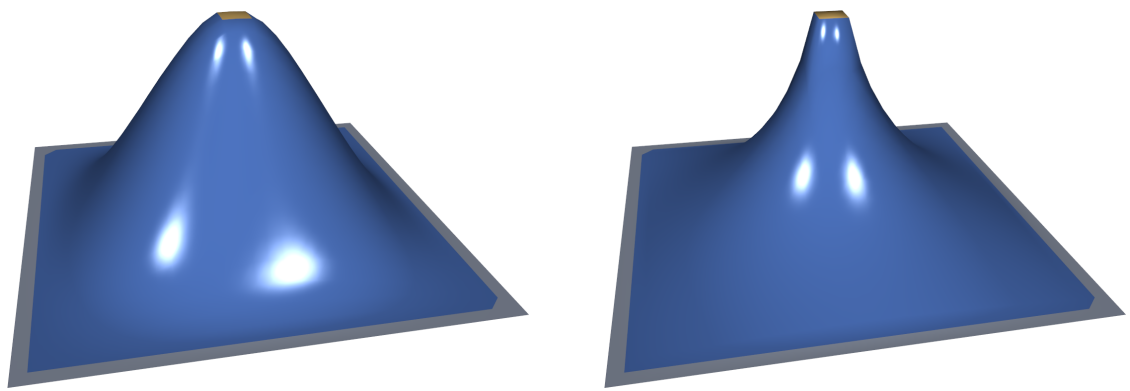


Volumetric Space Deformation

- Space-based discretization

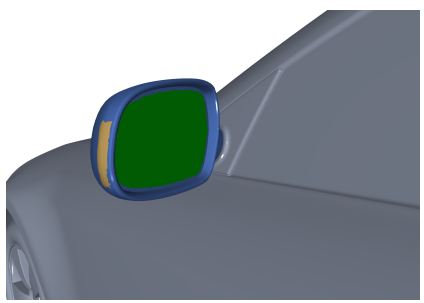


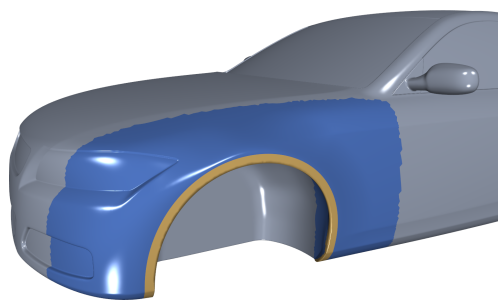
- Surface-based discretization

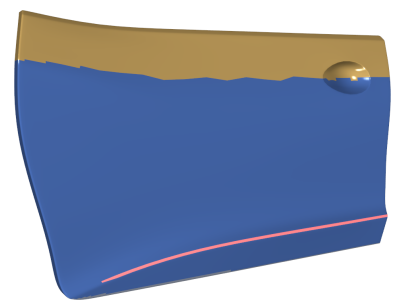


Geometric Constraints

- Design prototypes contain important geometric features
  - Planar components
  - Circular couplings or wheelhouses
  - Characteristic feature lines

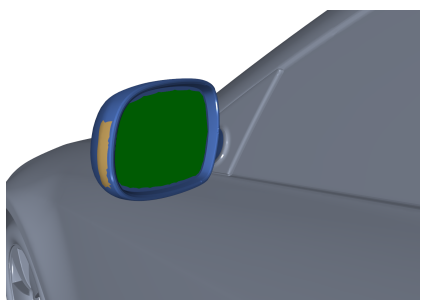


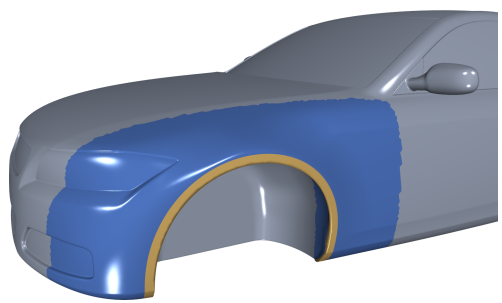


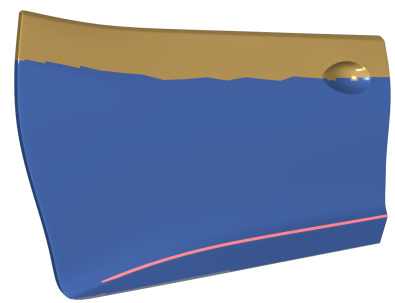


Geometric Constraints

- Design prototypes contain important geometric features
  - Planar components
  - Circular couplings or wheelhouses
  - Characteristic feature lines
- Deforming the design during optimization distorts features
  - Impaired functionality
  - Violated production limitations

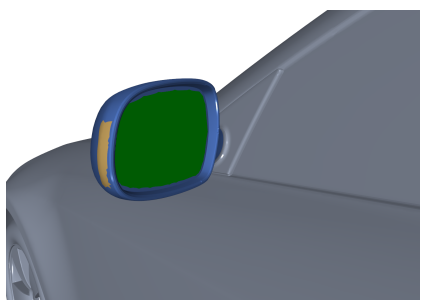
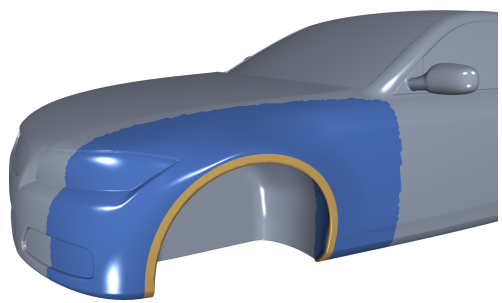
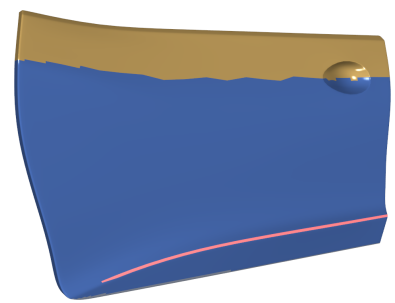






Geometric Constraints

- Design prototypes contain important geometric features
  - Planar components
  - Circular couplings or wheelhouses
  - Characteristic feature lines
- Deforming the design during optimization distorts features
  - Impaired functionality
  - Violated production limitations
- Classical solution: Add penalty terms to the cost function
  - *Creation* of infeasible designs
  - Costly *evaluation* (e.g., CFD)

Geometric Constraints

- Approach: Prevent distortion of features by incorporating geometric constraints into the deformation
  - + Only create feasible designs
  - + Avoid unnecessary performance evaluations

Geometric Constraints

- Approach: Prevent distortion of features by incorporating geometric constraints into the deformation
  - \+ Only create feasible designs
  - \+ Avoid unnecessary performance evaluations
- Constrained deformation techniques
  - − Most methods are surface-based

Geometric Constraints

- Approach: Prevent distortion of features by incorporating geometric constraints into the deformation
  - \+ Only create feasible designs
  - \+ Avoid unnecessary performance evaluations
- Constrained deformation techniques
  - − Most methods are surface-based
  - \+ Projection-based constraints[10]

10. Bouaziz et al., *Shape-Up: Shaping Discrete Geometry with Projections*, Computer Graphics Forum, 2012

Projection-Based Constraints

- Define projection operators P_t: Plane, circle, ...
- Measure deviation from prescribed constraints

$$E_{\text{constr}}(\boldsymbol{X}) \;=\; \sum_{t=1}^{m} \|\boldsymbol{X} - P_t(\boldsymbol{X})\|^2$$

Projection-Based Constraints

- Define projection operators P_t: Plane, circle, ...
- Measure deviation from prescribed constraints

$$E_{\text{constr}}(\boldsymbol{X}) = \sum_{t=1}^{m} \|\boldsymbol{X} - P_t(\boldsymbol{X})\|^2$$

- Projections P_t typically are nonlinear functions of $\boldsymbol{X}$

➔ Minimize E_{constr} by iterative alternating optimization

Projection-Based Constraints

- Define projection operators P_t: Plane, circle, ...
- Measure deviation from prescribed constraints

$$E_{\text{constr}}(\boldsymbol{X}) = \sum_{t=1}^{m} \|\boldsymbol{X} - P_t(\boldsymbol{X})\|^2$$

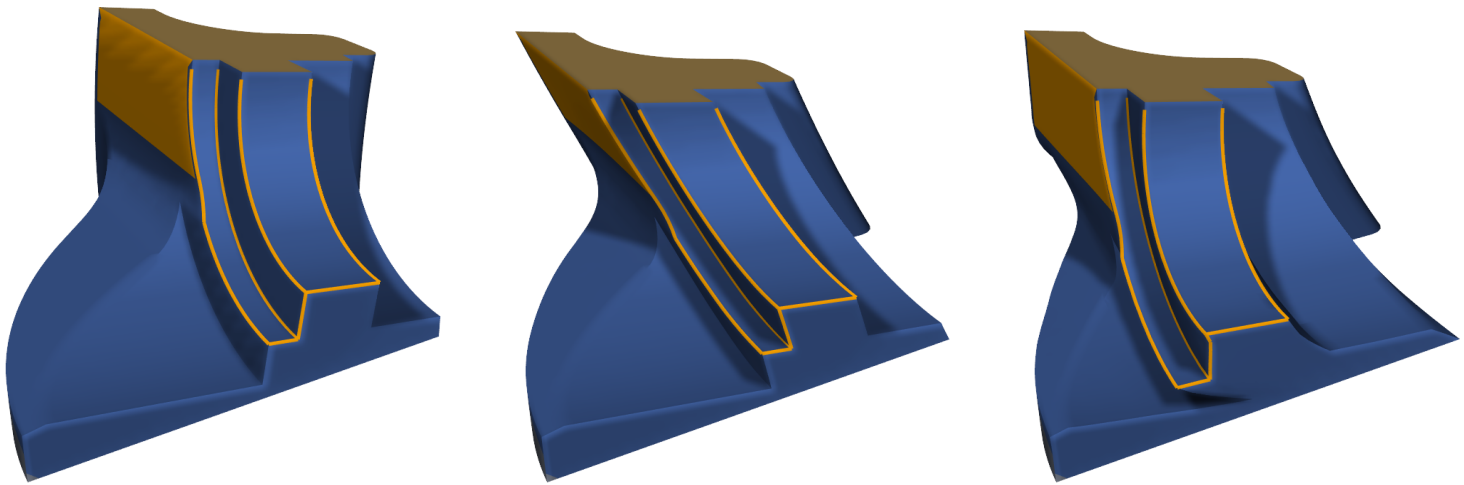
- Projections P_t typically are nonlinear functions of $\boldsymbol{X}$

➔ Minimize E_{constr} by iterative alternating optimization

- Integrated into deformation framework
- Extended original formulation to scale to large constraint areas

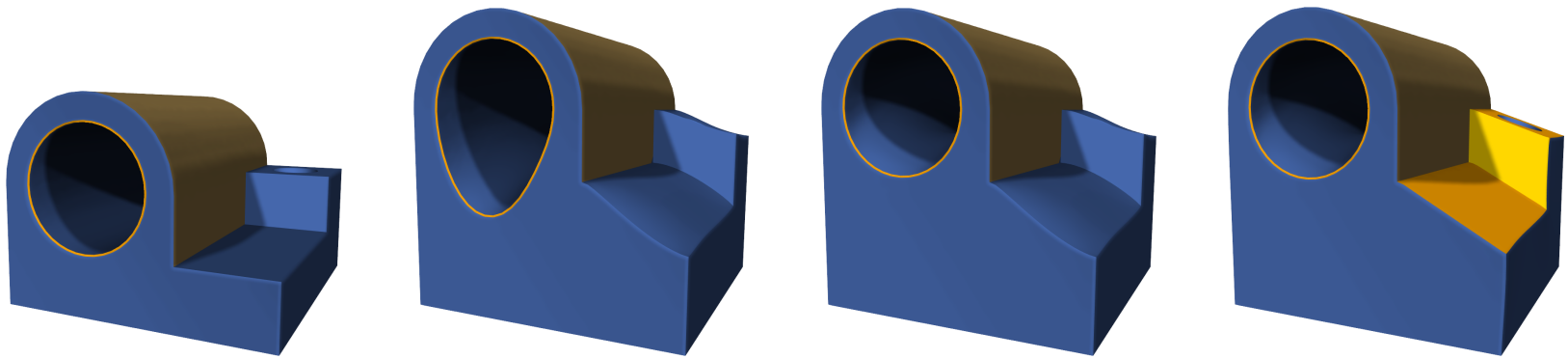
Geometric Constraint Examples

Fundamental geometric constraints: Planarity, circularity, feature lines



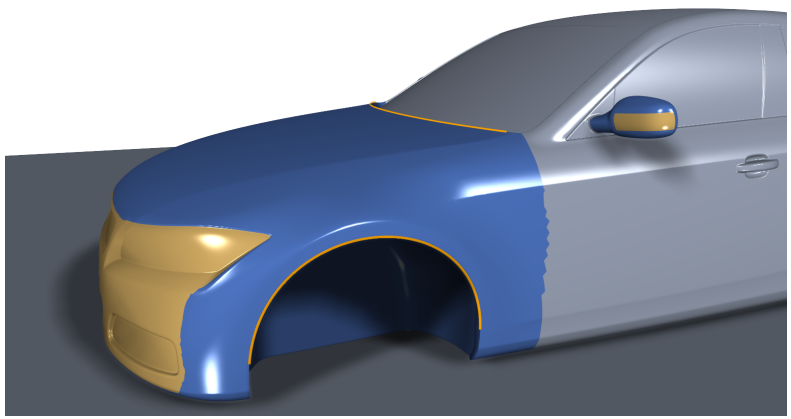
Geometric Constraint Examples

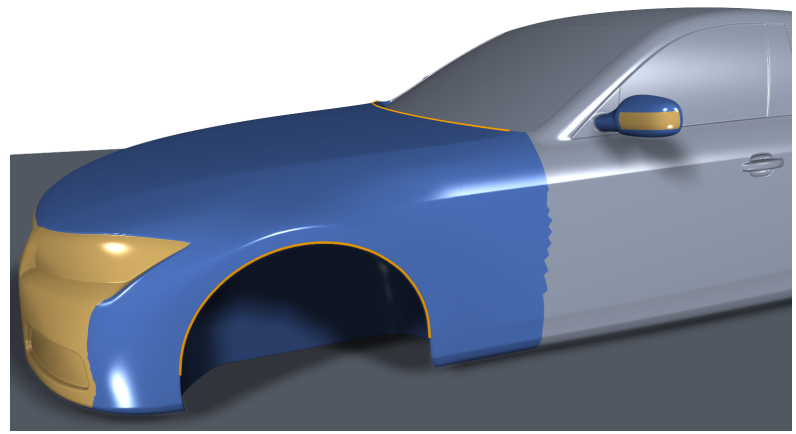
Fundamental geometric constraints: Planarity, circularity, feature lines



Geometric Constraint Examples

Fundamental geometric constraints: Planarity, circularity, feature lines

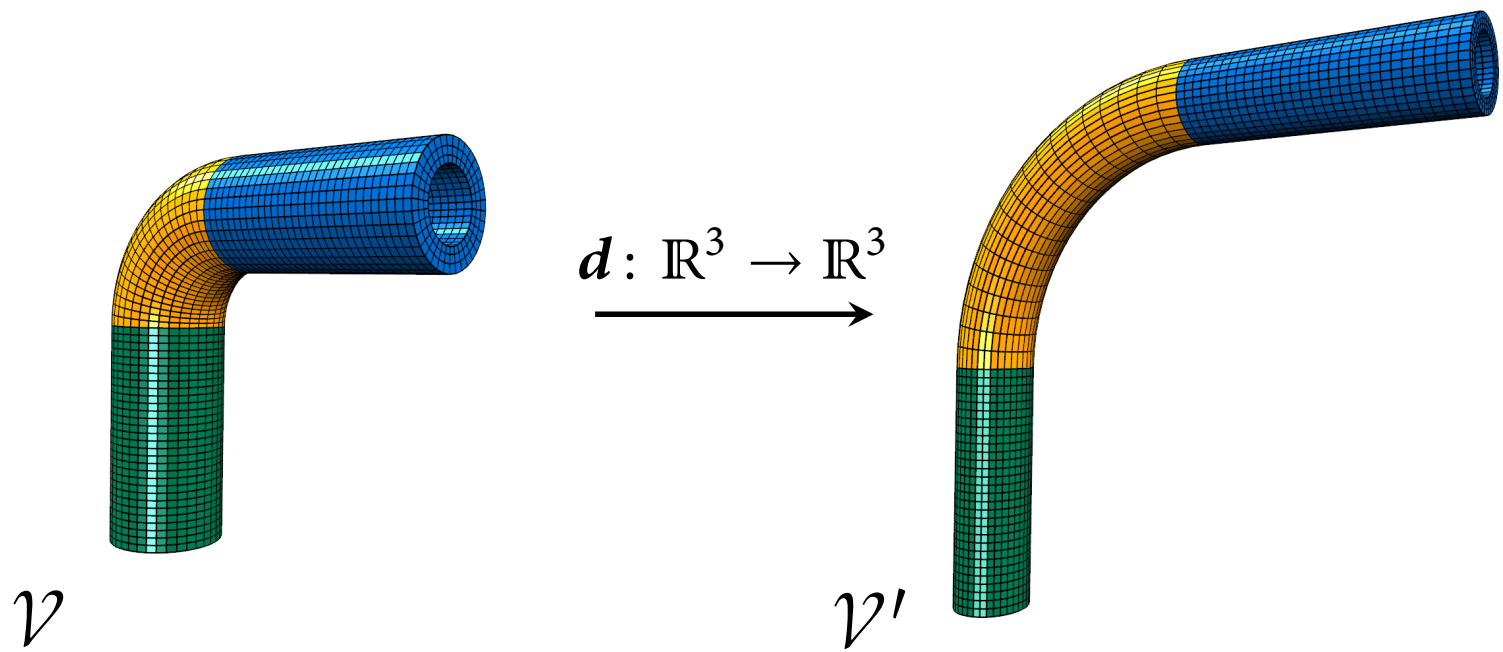




Volume Deformation Examples

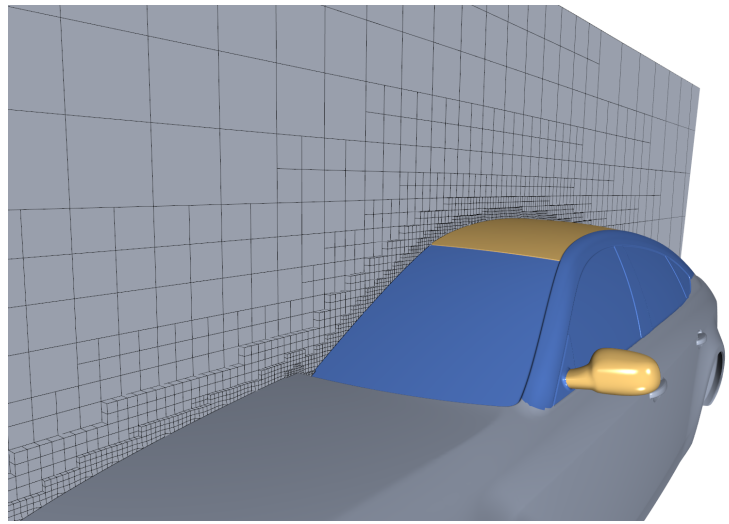
Comparison to previous results[11]

- Original mesh quality: 0.98
- After RBF deformation: 0.951
- Using our new method: 0.954

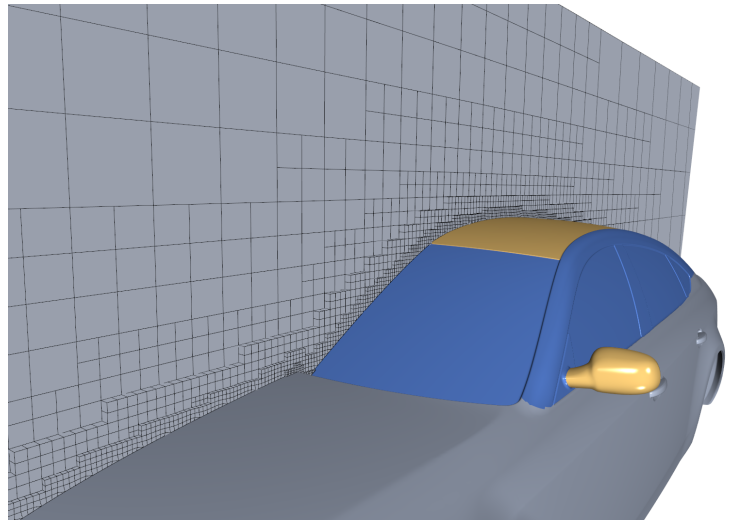


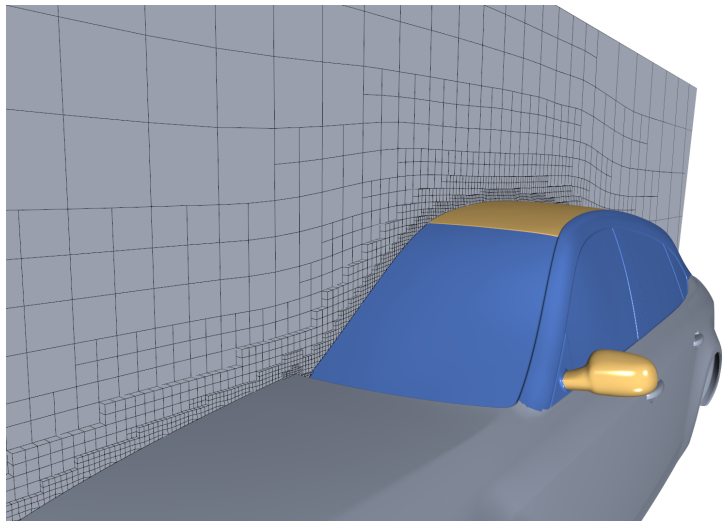
11. Sieger et al., *RBF Morphing Techniques for Simulation-based Design Optimization*, Engineering with Computers, 2014

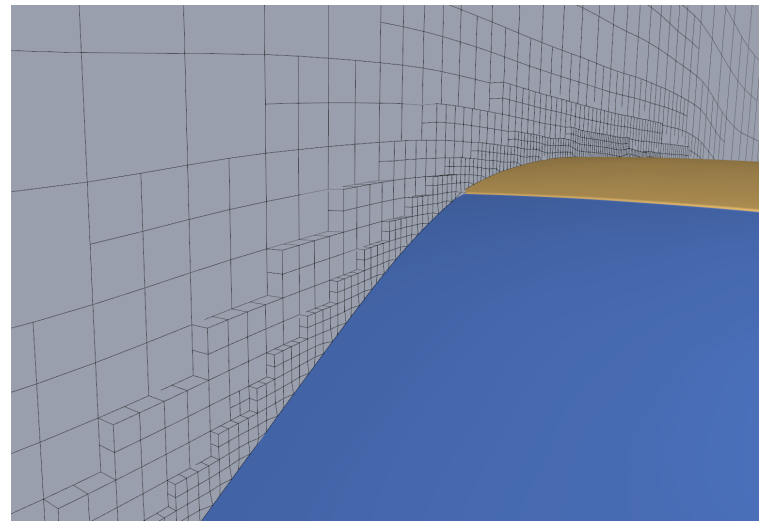
Combined Surface and Volume Deformation



Combined Surface and Volume Deformation







Constrained Deformation Summary

- A deformation technique for design optimization
  - + Modeling flexibility like surface-based methods
  - + Representation-independence of space deformations
  - + High quality comparable to RBFs
  - + Improved scalability through sparse linear systems
  - + Geometric constraints

Constrained Deformation Summary

- A deformation technique for design optimization
  - \+ Modeling flexibility like surface-based methods
  - \+ Representation-independence of space deformations
  - \+ High quality comparable to RBFs
  - \+ Improved scalability through sparse linear systems
  - \+ Geometric constraints
  - – Performance: Sampling, MLS basis functions, convergence
  - – Complexity: Implementation, parameter dependence

Constrained Deformation Summary

- A deformation technique for design optimization
  - \+ Modeling flexibility like surface-based methods
  - \+ Representation-independence of space deformations
  - \+ High quality comparable to RBFs
  - \+ Improved scalability through sparse linear systems
  - \+ Geometric constraints
  - – Performance: Sampling, MLS basis functions, convergence
  - – Complexity: Implementation, parameter dependence
- Central idea: Improve the design optimization process by integrating constraints *directly* into the deformation

Constrained Deformation Summary

- A deformation technique for design optimization
  - \+ Modeling flexibility like surface-based methods
  - \+ Representation-independence of space deformations
  - \+ High quality comparable to RBFs
  - \+ Improved scalability through sparse linear systems
  - \+ Geometric constraints
  - – Performance: Sampling, MLS basis functions, convergence
  - – Complexity: Implementation, parameter dependence
- Central idea: Improve the design optimization process by integrating constraints *directly* into the deformation
- Observation: Significant differences in the modeling flexibility and quality of RBFs and MLS

Directions for Future Work

- Additional constraint types, semantic relations
  - Mutual distances
  - Symmetry
  - Co-planarity
- More tight integration with CAD tools
  - Integration into CAD-based optimization framework
  - Transfer of constraints and functional requirements
- More extensive evaluation

Key Questions and Answers

1. What is a *good* deformation technique for design optimization?

2. How to *apply* and *improve* existing techniques?

3. How to incorporate *geometric constraints* into the deformation?

Key Questions and Answers

1. What is a *good* deformation technique for design optimization?

   Kernel-based space deformations provide the quality, flexibility, and robustness required for design optimization.

2. How to *apply* and *improve* existing techniques?

3. How to incorporate *geometric constraints* into the deformation?

Key Questions and Answers

1. What is a *good* deformation technique for design optimization?

   Kernel-based space deformations provide the quality, flexibility, and robustness required for design optimization.

2. How to *apply* and *improve* existing techniques?

   By *exploiting the flexibility of RBFs*, we can implement fully automatic design optimization loops based on CAD prototypes.

3. How to incorporate *geometric constraints* into the deformation?

Key Questions and Answers

1. What is a *good* deformation technique for design optimization?

   Kernel-based space deformations provide the quality, flexibility, and robustness required for design optimization.

2. How to *apply* and *improve* existing techniques?

   By *exploiting the flexibility of RBFs*, we can implement fully automatic design optimization loops based on CAD prototypes.

3. How to incorporate *geometric constraints* into the deformation?

   The combination of *MLS space deformations and projective constraints* allows for the creation of more feasible designs.

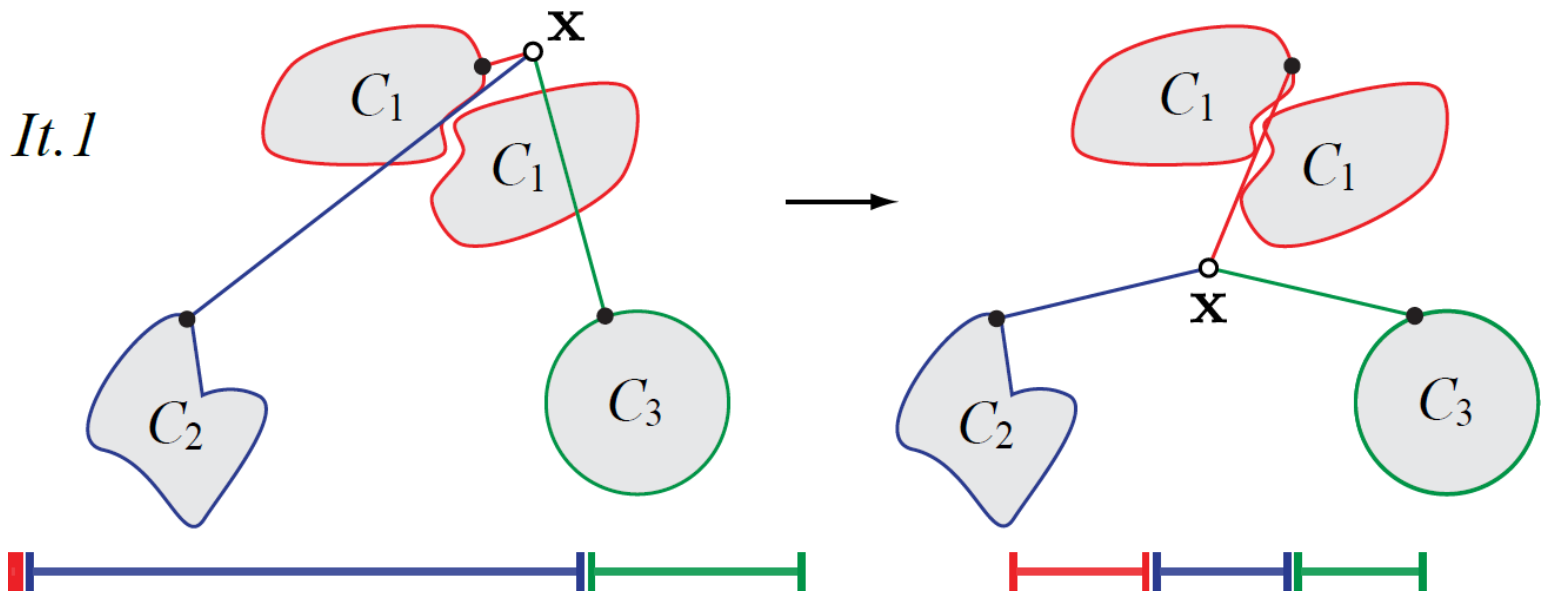
Acknowledgments

Thanks Mario and Stefan!

Thanks Honda Research Institute Europe

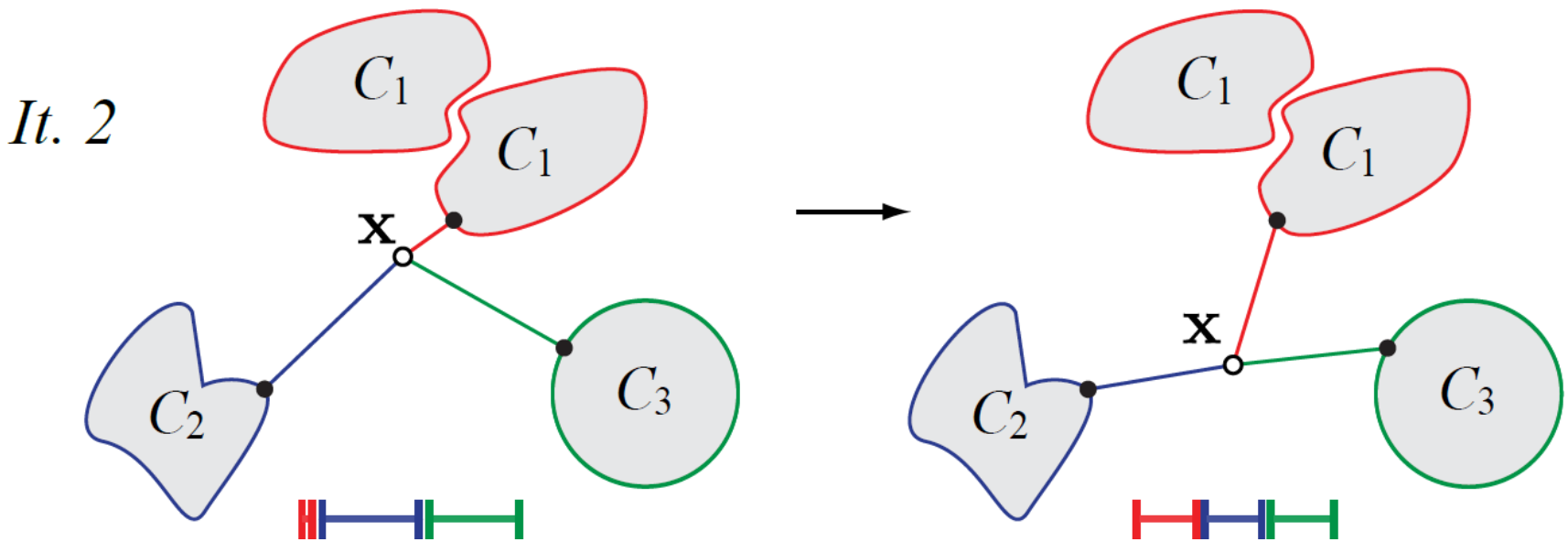
Projection-Based Constraints

- Alternating optimization:
  1. Project current point on constraint
  2. Solve linear system to minimize distance to constraints



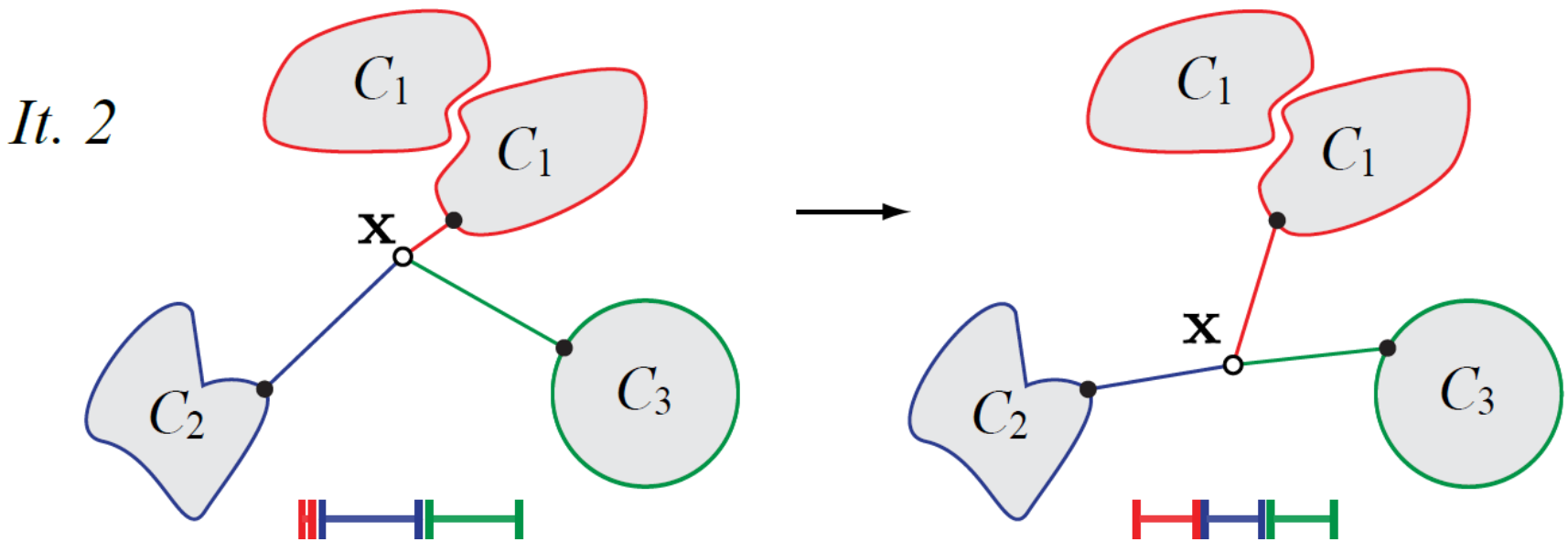
Projection-Based Constraints

- Alternating optimization:
  1. Project current point on constraint
  2. Solve linear system to minimize distance to constraints



Projection-Based Constraints

- Alternating optimization:
  1. Project current point on constraint
  2. Solve linear system to minimize distance to constraints
- \+ Flexible and simple formulation



Projection-Based Constraints

- Alternating optimization:
  1. Project current point on constraint
  2. Solve linear system to minimize distance to constraints

+ Flexible and simple formulation

− Slow convergence for complex constraints