Analysis-suitable Parameterization Techniques for Isogeometric Analysis

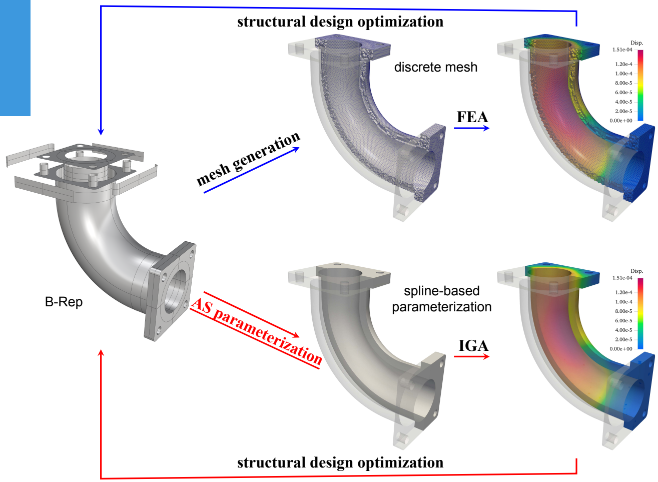
Delft University of Technology, the Netherlands

Ye Ji

11–13 Sep. 2023, Linz, Austria

Agenda

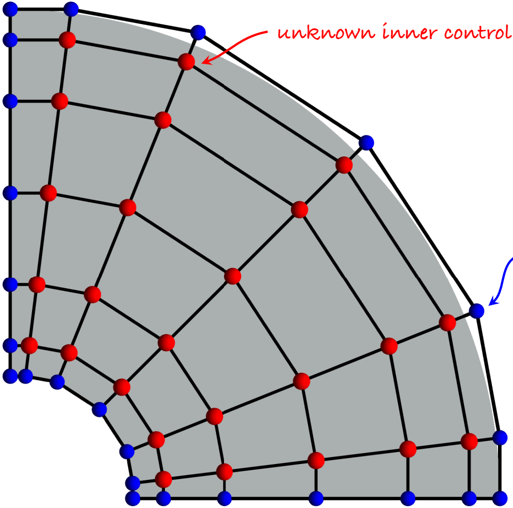
IsoGeometric Analysis (IGA)



Design-analysis-optimization product development workflow

- Introduced by T.J.R. Hughes et al., 2005.
- **Key Idea:** Approximate physical fields with the **same basis functions** used in CAD model representation.
- Benefits:
  - **Unified design and analysis;**
  - **Precise, efficient geometry;**
  - **No data type transition;**
  - **Simplified mesh refinement;**
  - **Continuous high-order fields;**
  - **Superior approximation.**
- Broad applications: shell analysis, fluid-structure interaction, shape and topology optimization, etc.

Problem Statement: Domain Parameterization



unknown inner control points $\mathbf{P}_i$

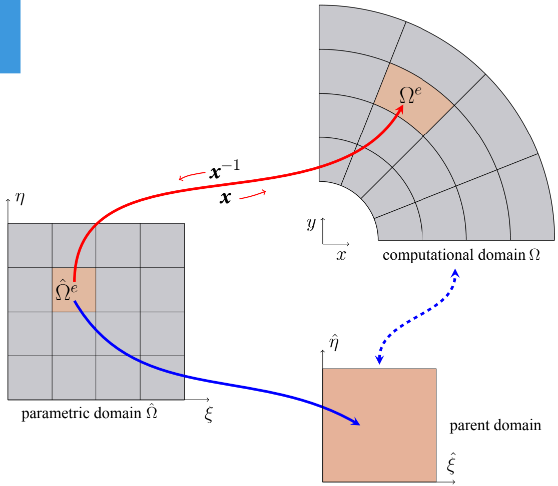
$$\boldsymbol{x}(\boldsymbol{\xi}) = \underbrace{\sum_{i\in\mathcal{I}_I} \mathbf{P}_i R_i(\boldsymbol{\xi})}_{\textbf{unknown}} + \underbrace{\sum_{j\in\mathcal{I}_B} \mathbf{P}_j R_j(\boldsymbol{\xi})}_{\textbf{known}}.$$

known boundary control points $\mathbf{P}_j$

GOAL: To construct the **unknown inner control points** $\mathbf{P}_i$ (or basis functions $R_i(\boldsymbol{\xi})$) such that $\boldsymbol{x}$ ensures **bijectivity** and exhibits minimal

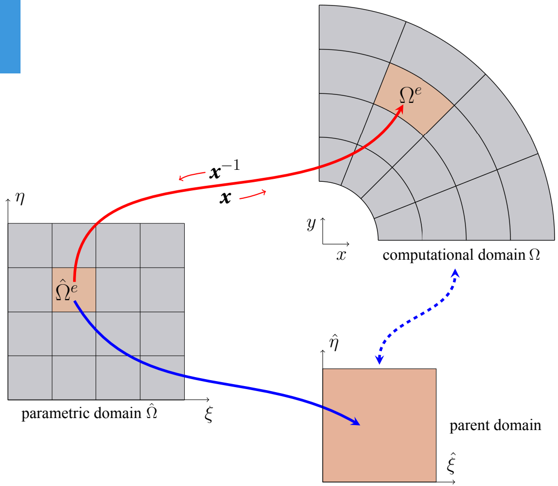
- **angle and volume distortion**;
- or **numerical error**.

Isotropic vs. Anisotroptic Parameterizations



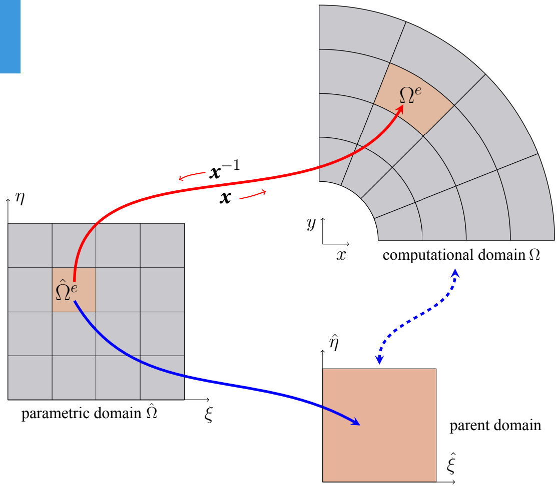
- **Bijectivity**: Why is it crucial?

Isotropic vs. Anisotroptic Parameterizations

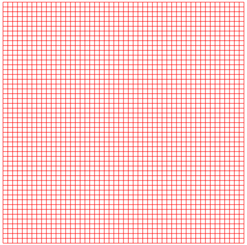


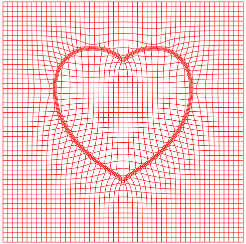
- **Bijectivity**: Why is it crucial?
- **Analysis-Suitable Parameterizations**
  - Should be **bijective** (non-unique solution);

Isotropic vs. Anisotroptic Parameterizations



- **Bijectivity**: Why is it crucial?
- **Analysis-Suitable Parameterizations**
  - Should be **bijective** (non-unique solution);
  - Minimize angle and area/volume distortion $\Rightarrow$ **Isotropic**;
  - or Minimize numerical error $\Rightarrow$ **Anisotropic**.





Agenda

Objective Function Terms

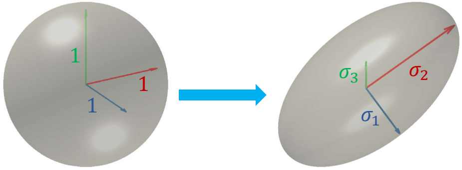
Angle Distortion

- Most Isometric ParametrizationS (MIPS) energy [HG2000, Fu+2015]:

$$\mathscr{E}^{\text{angle}} = \begin{cases} \frac{\sigma_1}{\sigma_2} + \frac{\sigma_2}{\sigma_1}, \\ \frac{1}{8}\left(\frac{\sigma_1}{\sigma_2} + \frac{\sigma_2}{\sigma_1}\right)\left(\frac{\sigma_2}{\sigma_3} + \frac{\sigma_3}{\sigma_2}\right)\left(\frac{\sigma_1}{\sigma_3} + \frac{\sigma_3}{\sigma_1}\right), \end{cases}$$

where σ_i are the singular values of $\mathcal{J}$.

- Ideally, $\sigma_1 = \sigma_2 = \cdots = \sigma_d$.

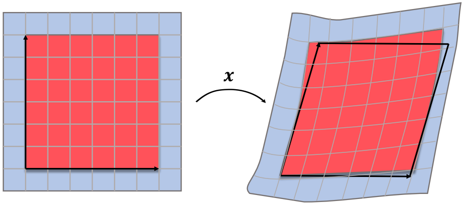


Area/Volume Distortion

- Area/volume distortion energy:

$$\mathscr{E}^{\text{unif.}}(\boldsymbol{x}) = \frac{|\mathcal{J}|}{vol(\Omega)} + \frac{vol(\Omega)}{|\mathcal{J}|},$$

where $vol(\Omega)$ denotes the area/volume of the computational domain Ω;



A Natural Method: Constrained Optimization Problem

- Solve the following constrained optimization problem, e.g., [Xu+2011, Pan+2020]:

$$\begin{aligned} \underset{\mathbf{P}_i,\ i\in\mathcal{I}_I}{\arg\min} \quad & \mathscr{E}(\boldsymbol{x}) = \int_{\hat{\Omega}} \left(\lambda_{\text{angle}} \mathscr{E}^{\text{angle}}(\boldsymbol{x}) \ + \lambda_{\text{unif.}} \mathscr{E}^{\text{unif.}}(\boldsymbol{x}) \right) \, \mathrm{d}\hat{\Omega}, \\ \text{s.t.} \quad & |\boldsymbol{\mathcal{J}}|_i > 0, \end{aligned}$$

  where $|\boldsymbol{\mathcal{J}}|_i$ are control coefficients of the Jacobian: $|\boldsymbol{\mathcal{J}}| = \sum_i |\boldsymbol{\mathcal{J}}|_i R_i(\boldsymbol{\xi})$.
- **Drawbacks**:
  - Hard to predict the lower bound of the Jacobian in advance;

A Natural Method: Constrained Optimization Problem

- Solve the following constrained optimization problem, e.g., [Xu+2011, Pan+2020]:

$$
\begin{aligned}
\underset{\mathbf{P}_i,\ i\in\mathcal{I}_I}{\arg\min} \quad & \mathscr{E}(\boldsymbol{x}) = \int_{\hat{\Omega}} \left(\lambda_{\text{angle}}\mathscr{E}^{\text{angle}}(\boldsymbol{x}) \ + \lambda_{\text{unif.}}\mathscr{E}^{\text{unif.}}(\boldsymbol{x})\right) \, \mathrm{d}\hat{\Omega}, \\
\text{s.t.} \quad & |\boldsymbol{\mathcal{J}}|_i > 0,
\end{aligned}
$$

  where $|\boldsymbol{\mathcal{J}}|_i$ are control coefficients of the Jacobian: $|\boldsymbol{\mathcal{J}}| = \sum_i |\boldsymbol{\mathcal{J}}|_i R_i(\boldsymbol{\xi})$.
- **Drawbacks**:
  - Hard to predict the lower bound of the Jacobian in advance;
  - **Constraint Complexity: Extremely large number of constraints**
    (*e.g., over **34k** for a* 20×20 *bi-cubic planar NURBS*) ◁ [Pan+2020, Ji+2021].

Equivalence Problem: Unconstrained Optimization

- Recall the planar MIPS energy,

$$\mathscr{E}_{2D}^{\text{angle}}(\boldsymbol{x}) = \frac{\sigma_1^2 + \sigma_2^2}{\sigma_1 \sigma_2} = \frac{trace(\boldsymbol{\mathcal{J}}^{\mathrm{T}}\boldsymbol{\mathcal{J}})}{|\boldsymbol{\mathcal{J}}|}.$$

  - MIPS energy **proceeds to infinity** if $|\boldsymbol{\mathcal{J}}| \to 0$.

Equivalence Problem: Unconstrained Optimization

- Recall the planar MIPS energy,

$$\mathscr{E}_{2D}^{\text{angle}}(\boldsymbol{x}) = \frac{\sigma_1^2 + \sigma_2^2}{\sigma_1 \sigma_2} = \frac{trace(\boldsymbol{\mathcal{J}}^{\mathrm{T}} \boldsymbol{\mathcal{J}})}{|\mathcal{J}|}.$$

  - MIPS energy **proceeds to infinity** if $|\mathcal{J}| \to 0$.
- Solve the following **unconstrained optimization problem**

$$\min_{\mathbf{P}_i,\ i \in \mathcal{I}_I} \quad \mathscr{E}(\boldsymbol{x}) = \int_{\hat{\Omega}} \left(\lambda_{\mathsf{angle}} \mathscr{E}^{\mathsf{angle}}(\boldsymbol{x}) + \lambda_{\mathsf{vol}} \mathscr{E}^{\mathsf{vol}}(\boldsymbol{x}) \right) \mathrm{d}\hat{\Omega}.$$

  - Avoid computation of control coefficients $|\mathcal{J}|_i$.

Equivalence Problem: Unconstrained Optimization

- Recall the planar MIPS energy,

$$\mathscr{E}_{2D}^{\text{angle}}(\boldsymbol{x}) = \frac{\sigma_1^2 + \sigma_2^2}{\sigma_1 \sigma_2} = \frac{trace(\boldsymbol{\mathcal{J}}^{\mathrm{T}} \boldsymbol{\mathcal{J}})}{|\boldsymbol{\mathcal{J}}|}.$$

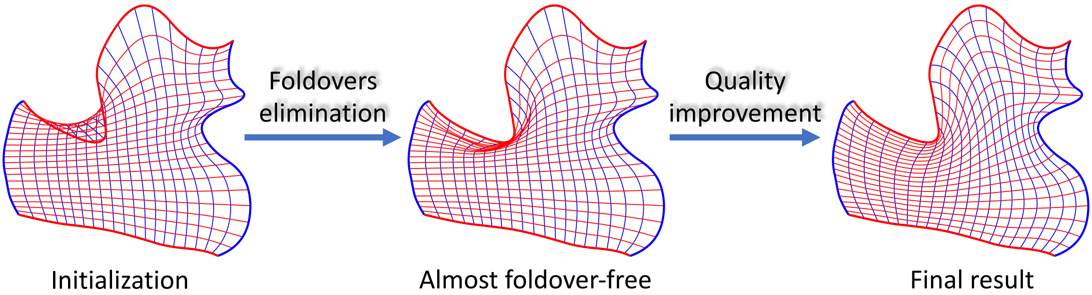
  - MIPS energy **proceeds to infinity** if $|\boldsymbol{\mathcal{J}}| \to 0$.
- Solve the following **unconstrained optimization problem**

$$\min_{\mathbf{P}_i,\ i \in \mathcal{I}_I} \quad \mathscr{E}(\boldsymbol{x}) = \int_{\hat{\Omega}} \left(\lambda_{\mathsf{angle}} \mathscr{E}^{\mathsf{angle}}(\boldsymbol{x}) + \lambda_{\mathsf{vol}} \mathscr{E}^{\mathsf{vol}}(\boldsymbol{x}) \right) \mathrm{d}\hat{\Omega}.$$

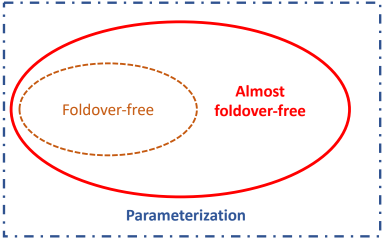
  - Avoid computation of control coefficients $|\boldsymbol{\mathcal{J}}|_i$.
  - **Prerequisite**: An already bijective initialization is needed.

Barrier Function-based Parameterization Construction

- **Step I: Initialization**
  - Initialize using an algebraic method, such as the Coons Patch.
- **Step II: Foldovers Elimination**
  - Employ techniques to remove foldovers in the parameterization.
- **Step III: Quality Improvement**
  - Refine the parameterization to enhance its quality.



Foldovers Elimination: Almost Foldover-Free

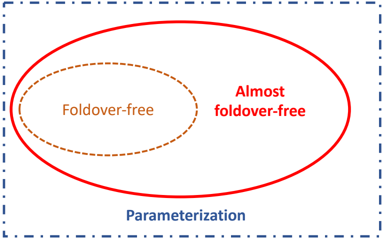


- Solve the following optimization problem:

$$\underset{\mathbf{P}_i,\ i\in\mathcal{I}_I}{\arg\min} \quad E(\boldsymbol{x}) = \int_{\hat{\Omega}} \max\left(0, \delta - |\mathcal{J}|\right)\ \mathrm{d}\hat{\Omega},$$

where δ is a threshold ($\delta = 5\% vol(\Omega)$ as default).

Foldovers Elimination: Almost Foldover-Free



- Solve the following optimization problem:

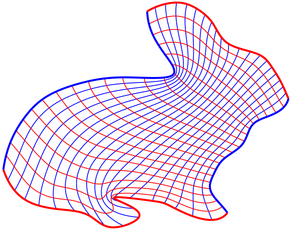
$$\underset{\mathbf{P}_i,\ i\in\mathcal{I}_I}{\arg\min} \quad E(\boldsymbol{x}) = \int_{\hat{\Omega}} \max\left(0, \delta - |\mathcal{J}|\right) \,\mathrm{d}\hat{\Omega},$$

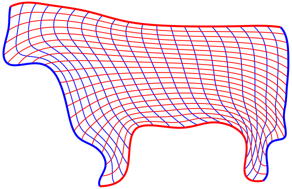
  where δ is a threshold ($\delta = 5\% vol(\Omega)$ as default).
- Quality improvement (robustness consideration):

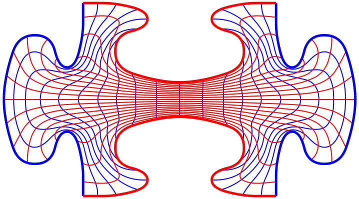
$$\mathscr{E}^c = \begin{cases} \int_{\hat{\Omega}} \left(\lambda_{\text{angle}}\mathscr{E}^{\text{angle}}(\boldsymbol{x}) + \lambda_{\text{vol}}\mathscr{E}^{\text{vol}}(\boldsymbol{x})\right) \,\mathrm{d}\hat{\Omega}, & \min|\mathcal{J}| > 0, \\ +\infty, & \text{otherwise.} \end{cases}$$

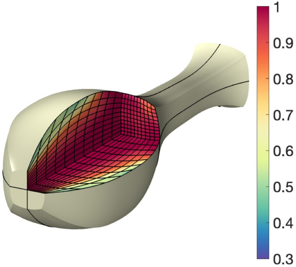
- Analytical gradient
  - for numerical stability and computational efficiency.

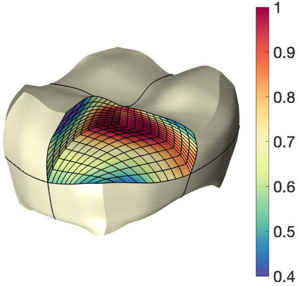
Gallery: Barrier Function-based Method

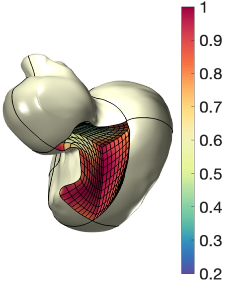






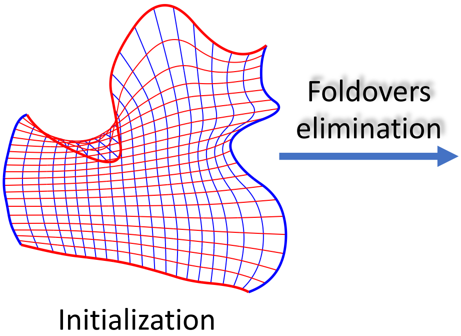






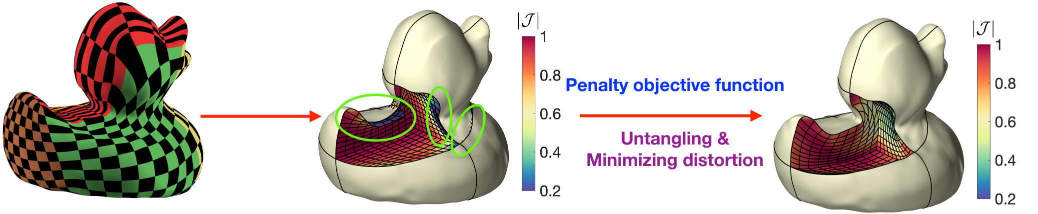
Penalty Function-based Parameterization Construction

- **Foldover elimination is often necessary but does not inherently improve parameterization quality.**
- Is it feasible to skip the foldover elimination step?



Penalty Function-based Parameterization Construction

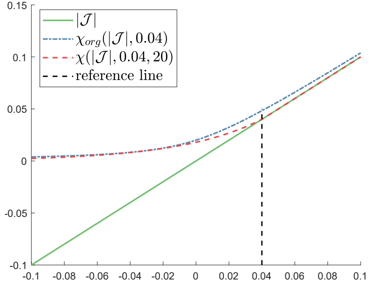
- **Foldover elimination is often necessary but does not inherently improve parameterization quality.**
- Is it feasible to skip the foldover elimination step?

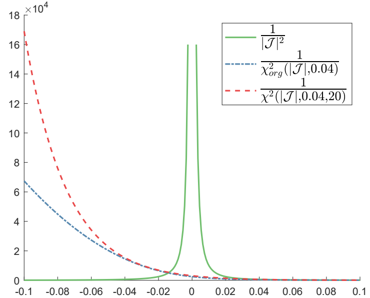


B-Rep
Initialization
Optimized parameterization

- **Certainly! Simultaneously untangling and minimizing distortion!!!**

Basic Idea: Penalty Function





- **Penalty function:**

$$\chi(|\mathcal{J}|, \varepsilon, \beta) = \begin{cases} \varepsilon \cdot e^{\beta(|\mathcal{J}|-\varepsilon)} & \text{if } |\mathcal{J}| \leq \varepsilon \\ |\mathcal{J}| & \text{if } |\mathcal{J}| > \varepsilon \end{cases},$$

  where $\varepsilon > 0$ is small, and β is a penalty factor;

- $\chi(|\mathcal{J}|, \varepsilon, \beta)$ approaches ε for small $|\mathcal{J}|$, and equals $|\mathcal{J}|$ otherwise.

- $\frac{1}{\chi^2(|\mathcal{J}|,\varepsilon,\beta)}$:
  - **Penalizes negative Jacobians** with large values
  - **Accepts positive Jacobians** with small values

Jacobian Regularization & Revised Objective Function

- **Only one optimization problem needs to be solved:**

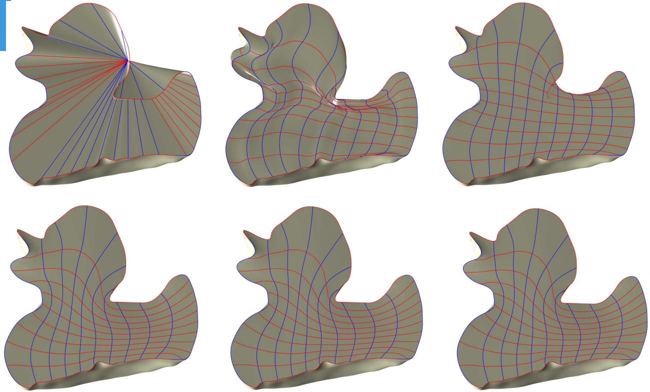
$$\underset{\mathbf{P}_i,\ i\in\mathcal{I}_I}{\arg\min}\, \mathscr{E}^c = \int_{\hat{\Omega}} \left(\lambda_{\text{mips}} \mathscr{E}^c_{\text{mips}} + \lambda_{\text{vol}} \mathscr{E}^c_{\text{vol}}\right)\, \mathrm{d}\hat{\Omega}$$

$$= \int_{\hat{\Omega}} \left(\frac{\lambda_{\text{mips}}}{8} \kappa_F^2(\boldsymbol{\mathcal{J}}) \cdot \frac{|\boldsymbol{\mathcal{J}}|^2}{\chi^2(|\boldsymbol{\mathcal{J}}|, \varepsilon, \beta)} + \lambda_{\text{vol}} \left(\frac{vol(\Omega)}{\chi(|\boldsymbol{\mathcal{J}}|, \varepsilon, \beta)} + \frac{\chi(|\boldsymbol{\mathcal{J}}|, \varepsilon, \beta)}{vol(\Omega)} \right) \right)\, \mathrm{d}\hat{\Omega},$$

where $\mathbf{P}_i,\ i \in \mathcal{I}_I$ are the unknown inner control points.

- **Computational Techniques:**
  - Analytical gradient for numerical stability and efficiency;
  - Reduced numerical integration complexity;
  - Pre-computation for faster calculations.

Different Initializations & Stress Test

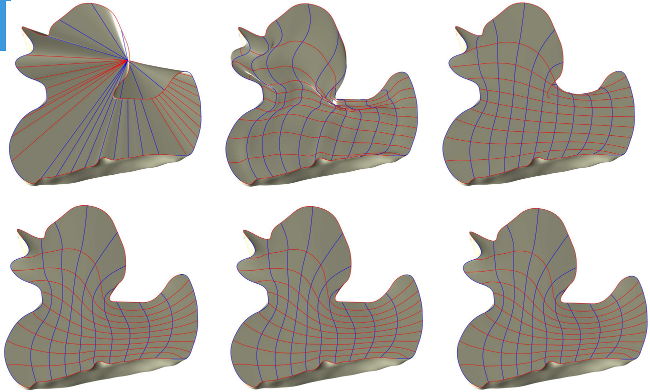


Same point Discrete Coons Smoothness energy

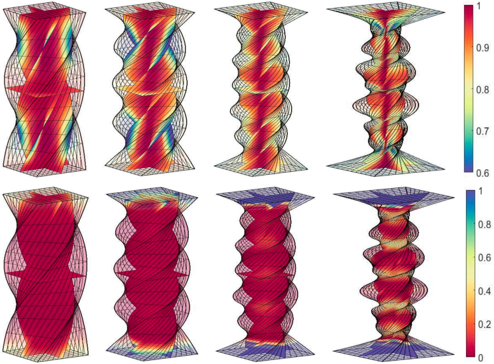
- The results show strong similarity.
- Insensitive to varying initializations.

Different Initializations & Stress Test

- The results show strong similarity.
- Insensitive to varying initializations.

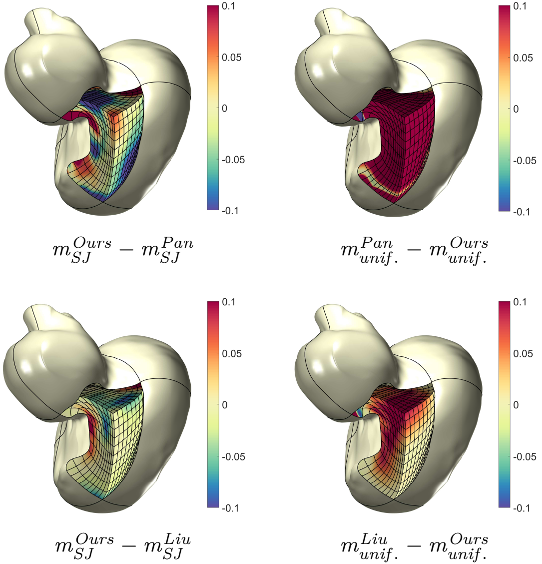


Same point | Discrete Coons | Smoothness energy



Rot180 | Rot360 | Rot540 | Rot720

Parameterization Quality Comparison



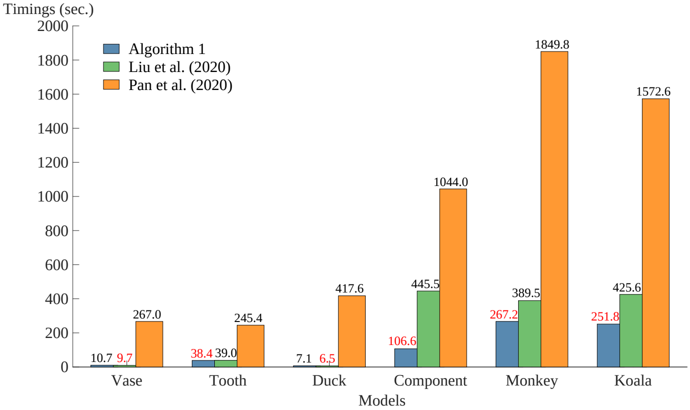
- Comparisons to Pan et al. 2020 [a] and Liu et al. 2020 [b].
- Positive values (red regions) suggest our method performs better in terms of both angle and volume distortion.

[a] Pan, M., Chen, F., & Tong, W. (2020). Volumetric spline parameterization for isogeometric analysis. Computer Methods in Applied Mechanics and Engineering, 359, 112769.

[b] Liu, H., Yang, Y., Liu, Y., & Fu, X. M. (2020). Simultaneous interior and boundary optimization of volumetric domain parameterizations for IGA. Computer Aided Geometric Design, 79, 101853.

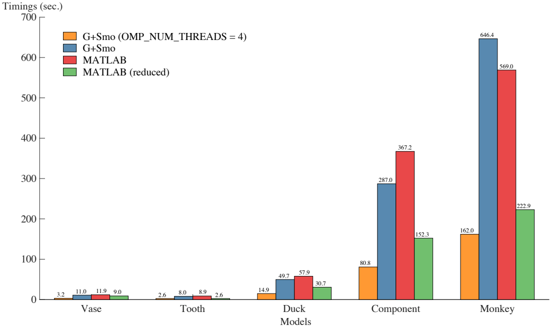
Efficiency Comparison

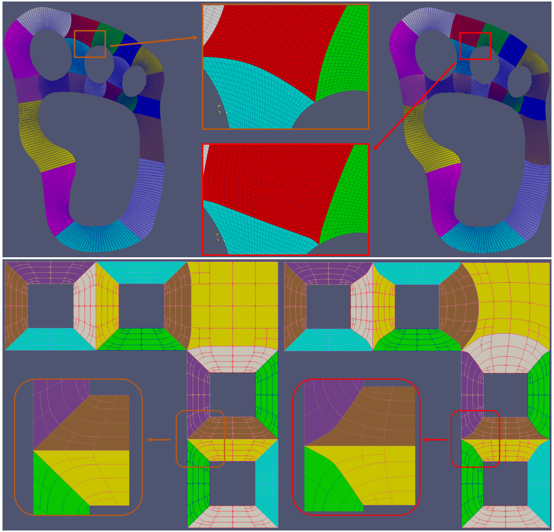
- Significantly faster than Pan et al. (2020) due to no constraints.
- **Large-Scale Models:** Outperforms Liu et al. (2020).
- **Implementation:** Speed comparisons may vary
  - Our use of MATLAB vs. C++ for competitors.



G+Smo Implementation

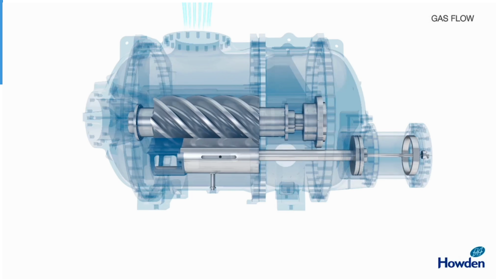
- In our released `G+Smo` implementation, $3-4$x **speed-up**;
- **Fast matrix assembly in progress**.





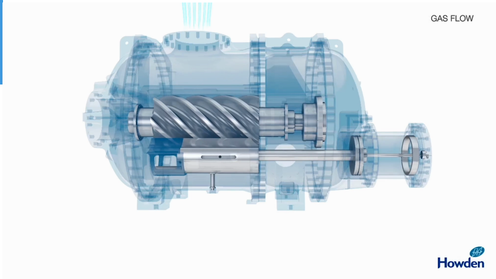
Agenda

Rotary twin-screw machines

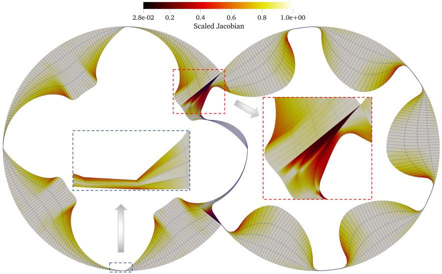


Rotary twin-screw compressor source

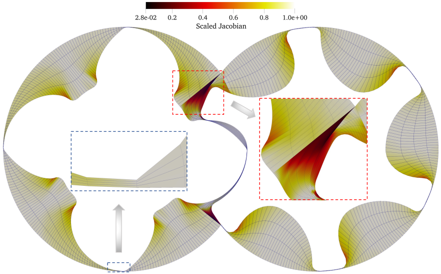
Rotary twin-screw machines



Rotary twin-screw compressor source



Penalty function-based method [a]



Elliptic Grid Generation method [b]

[a] Ref.: Ji, Y. et al. (2022). Penalty function-based volumetric... Computer Aided Geometric Design, 94, 102081.

[b] Ref.: Hinz, J.P. et al. (2018). Elliptic grid generation techniques... Computer Aided Geometric Design, 65, 48-75.

Elliptic Grid Generation (EGG) method

- To compute a harmonic mapping $\boldsymbol{x} : \hat{\Omega} \to \Omega$ by solving Laplace equations:

$$\begin{cases} \nabla \cdot \nabla \xi(x, y) = 0 \\ \nabla \cdot \nabla \eta(x, y) = 0 \end{cases} \quad \text{s.t. } \boldsymbol{x}^{-1}|_{\partial\Omega} = \partial\hat{\Omega}.$$

[1] Eells, J., & Lemaire, L., (1978). A report on harmonic maps. Bulletin of the London mathematical society, 10(1):1–68.

[2] Duren, P., & Hengartner, W., (1997). Harmonic mappings of multiply connected domains. Pac. J. Math. 180, 201–220.

Elliptic Grid Generation (EGG) method

- To compute a harmonic mapping $\boldsymbol{x}: \hat{\Omega} \to \Omega$ by solving Laplace equations:

$$\begin{cases} \nabla \cdot \nabla \xi(x, y) = 0 \\ \nabla \cdot \nabla \eta(x, y) = 0 \end{cases} \quad \text{s.t. } \boldsymbol{x}^{-1}|_{\partial\Omega} = \partial\hat{\Omega}.$$

- The **existence and uniqueness** of the harmonic mapping $\boldsymbol{x}^{-1}$ is guaranteed if [a]:
  - The curvature of $\hat{\Omega}$ is non-positive;
  - The boundary $\hat{\Omega}$, when considered with respect to the metric on Ω, is convex.

[1] Eells, J., & Lemaire, L., (1978). A report on harmonic maps. Bulletin of the London mathematical society, 10(1):1–68.
[2] Duren, P., & Hengartner, W., (1997). Harmonic mappings of multiply connected domains. Pac. J. Math. 180, 201–220.

Elliptic Grid Generation (EGG) method

- To compute a harmonic mapping $\boldsymbol{x}: \hat{\Omega} \to \Omega$ by solving Laplace equations:

$$\begin{cases} \nabla \cdot \nabla \xi(x, y) = 0 \\ \nabla \cdot \nabla \eta(x, y) = 0 \end{cases} \quad \text{s.t. } \boldsymbol{x}^{-1}|_{\partial\Omega} = \partial\hat{\Omega}.$$

- The **existence and uniqueness** of the harmonic mapping $\boldsymbol{x}^{-1}$ is guaranteed if [a]:
  - The curvature of $\hat{\Omega}$ is non-positive;
  - The boundary $\hat{\Omega}$, when considered with respect to the metric on Ω, is convex.
- The unique solution $\boldsymbol{x}^{-1}$ offers a **one-to-one mapping** (with the Jacobian $\boldsymbol{J}$ not vanishing), which is ensured by the Radó-Kneser-Choquet theorem. [b]

[1] Eells, J., & Lemaire, L., (1978). A report on harmonic maps. Bulletin of the London mathematical society, 10(1):1–68.
[2] Duren, P., & Hengartner, W., (1997). Harmonic mappings of multiply connected domains. Pac. J. Math. 180, 201–220.

Elliptic Grid Generation (EGG) & Its H^2 Discretization

- To compute a harmonic mapping $\boldsymbol{x}: \hat{\Omega} \to \Omega$ by solving Laplace equations:

$$\begin{cases} \nabla \cdot \nabla \xi(x, y) = 0 \\ \nabla \cdot \nabla \eta(x, y) = 0 \end{cases} \quad \text{s.t. } \boldsymbol{x}^{-1}|_{\partial\Omega} = \partial\hat{\Omega}.$$

[1] Hinz, J. P. (2020). PDE-Based Parameterization Techniques for Isogeometric Analysis Applications.

Elliptic Grid Generation (EGG) & Its H^2 Discretization

- To compute a harmonic mapping $\boldsymbol{x}: \hat{\Omega} \to \Omega$ by solving Laplace equations:

$$\begin{cases} \nabla \cdot \nabla \xi(x, y) = 0 \\ \nabla \cdot \nabla \eta(x, y) = 0 \end{cases} \quad \text{s.t. } \boldsymbol{x}^{-1}|_{\partial\Omega} = \partial\hat{\Omega}.$$

- Nonlinear vector-valued second-order PDE [1]

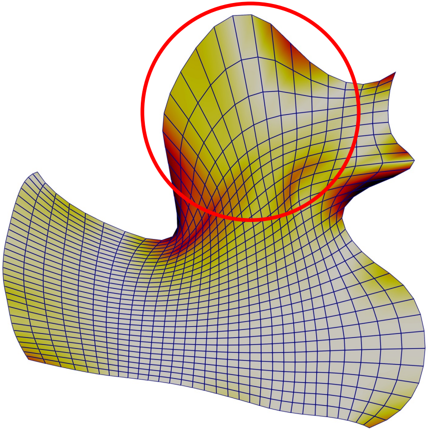
$$\forall R_i \in \Sigma_0 : \begin{cases} \int_{\hat{\Omega}} \mathbf{R}\ \widetilde{\mathcal{L}} x \ d\hat{\Omega} = \mathbf{0}, \\ \int_{\hat{\Omega}} \mathbf{R}\ \widetilde{\mathcal{L}} y \ d\hat{\Omega} = \mathbf{0}, \end{cases} \quad \text{s.t. } \boldsymbol{x}|_{\partial\hat{\Omega}} = \partial\Omega,$$

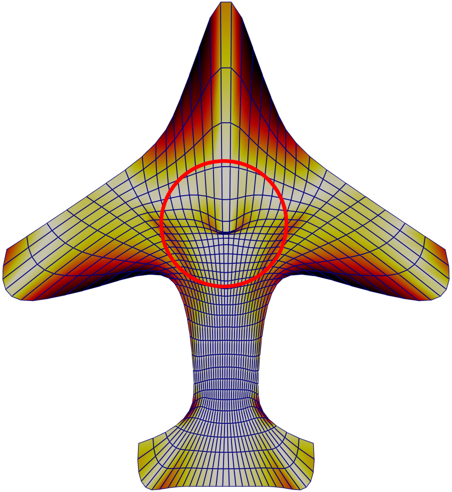
where

$$\widetilde{\mathcal{L}} = \frac{\mathcal{L}}{g_{11} + g_{22}}, \quad \text{and } \mathcal{L} = g_{22} \frac{\partial^2}{\partial \xi^2} - 2 g_{12} \frac{\partial^2}{\partial \xi \partial \eta} + g_{11} \frac{\partial^2}{\partial \eta^2}.$$

[1] Hinz, J. P. (2020). PDE-Based Parameterization Techniques for Isogeometric Analysis Applications.

Elliptic Grid Generation (EGG) & Its H^2 Discretization





- Non-uniform elements appear, may even result in non-bijective results.

[1] Hinz, J. P. (2020). PDE-Based Parameterization Techniques for Isogeometric Analysis Applications.

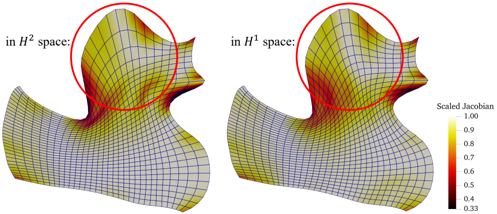
Quasi-Hamonic Mapping

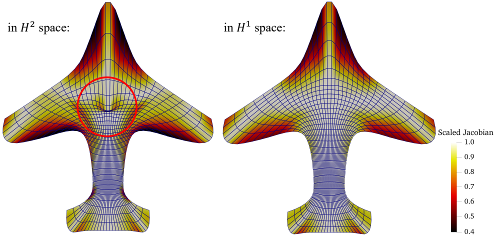
- We incorporate a monitor function $\mathbb{A}$ into our governing equation [1]:

$$-\nabla \cdot (\mathbb{A} \nabla \boldsymbol{\xi}) = \mathbf{0}.$$

where the monitor function is defined as

$$\mathbb{A} = \begin{bmatrix} \frac{1}{|\mathcal{J}|} & 0 \\ 0 & \frac{1}{|\mathcal{J}|} \end{bmatrix}.$$





[1] Thanks to a discussion with Jochen Peter Hinz in Genoa, Jul. 2023.

Discretization in Sobolev space H^1

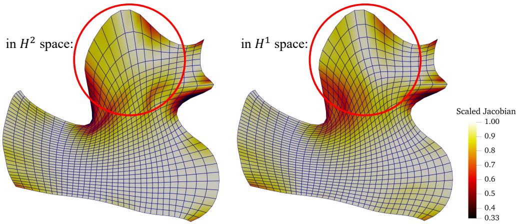
- The variational formulation in the Sobolev space H^1 reads

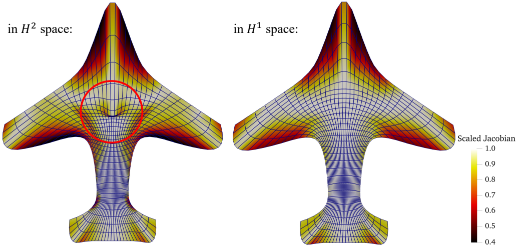
$$\forall R_i \in \Sigma_0 : \begin{cases} \mathbf{F}^x_{H^1} = \mathbf{0}, \\ \mathbf{F}^y_{H^1} = \mathbf{0}, \end{cases} \quad \text{s.t. } \mathbf{x}|_{\partial\hat{\Omega}} = \partial\Omega.$$

where

$$\mathbf{F}^x_{H^1} = \int_{\hat{\Omega}} \nabla_{\mathbf{x}}\mathbf{R} \cdot \nabla_{\mathbf{x}}\xi \; d\hat{\Omega}, \quad \mathbf{F}^y_{H^1} = \int_{\hat{\Omega}} \nabla_{\mathbf{x}}\mathbf{R} \cdot \nabla_{\mathbf{x}}\eta \; d\hat{\Omega},$$

and $\mathbf{R}$ denotes the column collection of the NURBS basis functions $R_i \in \Sigma_0$.

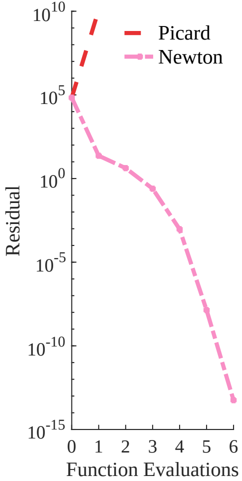




Picard Iteration

Bratu problem (64×64):

$$\Delta u + \lambda e^u = 0, \quad in \quad \Omega,$$
$$u = 0, \quad on \quad \partial\Omega.$$



Nonlinear system $\iff$ Fixed-point iteration :

$$\mathcal{F}(\boldsymbol{u}) = 0 \iff \boldsymbol{u} = \mathcal{G}(\boldsymbol{u}) = \boldsymbol{u} + \mathcal{F}(\boldsymbol{u})$$

for some $\mathcal{F} : \mathbb{R}^n \to \mathbb{R}^n$ and $\mathcal{G} : \mathbb{R}^n \to \mathbb{R}^n$.

- Basic Fixed-Point Iteration:

Algorithm Picard: Picard Iteration

```
1 Given u_0.
2 for k = 1, 2, ..., itmax until ||F_k|| < tol do
3   Set u_{k+1} = G(u^k)
```

- **Converges too slowly to be useful, may even diverge.**

Anderson Acceleration (D.G. Anderson, 1965)

- AA(m) linearly recombines m **previous iterates** in a manner that approximately **minimizes the linearized fixed-point residual $\mathcal{F}$ in a least-squares fashion**.

Algorithm AA: Anderson Acceleration

Given $\boldsymbol{u}_0$ and window size $m \geq 1$;

Set $\boldsymbol{u}_1 = \mathcal{G}(\boldsymbol{u}_0)$;

for $k = 1, 2, \ldots, itmax$ *until* $\|\mathcal{F}_k\| < tol$ **do**

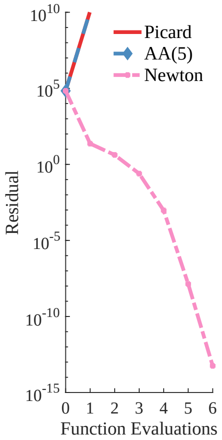
  Set $m_k = \min\{m, k\}$;

  Determine $\boldsymbol{\alpha}^{(k)} = \left(\alpha_0^{(k)}, \alpha_1^{(k)}, \ldots, \alpha_{m_k}^{(k)}\right)^{\mathrm{T}}$ that solves

$$\underset{\boldsymbol{\alpha}=(\alpha_0,\alpha_1,\ldots,\alpha_{m_k})^{\mathrm{T}}}{\arg\min} \left\| \sum_{i=0}^{m_k} \mathcal{F}_{k-m_k+i}\boldsymbol{\alpha} \right\|_2^2, \quad \text{s.t.} \quad \sum_{i=0}^{m_k} \alpha_i = 1;$$

  Update $\boldsymbol{u}_{k+1} = \sum_{i=0}^{m_k} \alpha_i^{(k)} \mathcal{G}_{k-m_k+i}$;

Bratu problem (64×64):



Enhancing Anderson Acceleration with Preconditioning

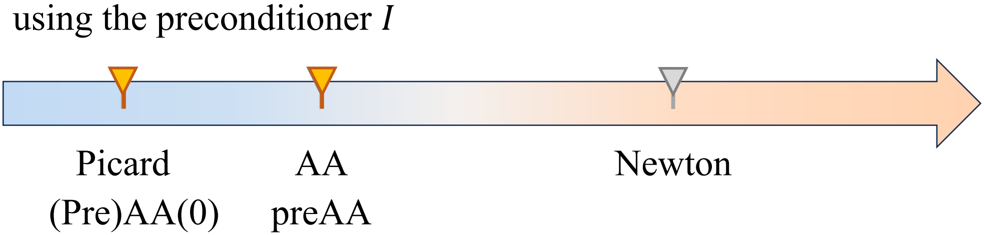
Preconditioning Strategy

We introduce a preconditioning strategy for the fixed-point iteration scheme:

$$\boldsymbol{u}_{k+1} = \boldsymbol{u}_k - \boldsymbol{\mathcal{M}_k}^{-1} \mathcal{F}_k,$$

where $\boldsymbol{\mathcal{M}}_k$ is a non-singular matrix, known as the **preconditioner** at iteration k.

- By setting $\boldsymbol{\mathcal{M}}_k$ to the **identity matrix**, PreAA degenerates to the original AA.
- AA(0) is essentially the same as Picard iteration.



Enhancing Anderson Acceleration with Preconditioning

Preconditioning Strategy

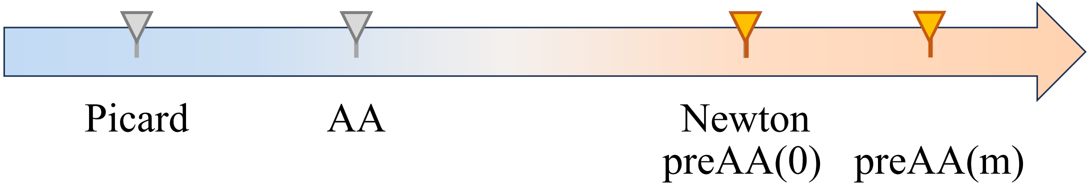
We introduce a preconditioning strategy for the fixed-point iteration scheme:

$$\boldsymbol{u}_{k+1} = \boldsymbol{u}_k - {\boldsymbol{\mathcal{M}}_k}^{-1} \boldsymbol{\mathcal{F}}_k,$$

where $\boldsymbol{\mathcal{M}}_k$ is a non-singular matrix, known as the **preconditioner** at iteration k.

- By setting $\boldsymbol{\mathcal{M}}_k$ to $jac(\boldsymbol{\mathcal{F}})$, PreAA(0) degenerates to Newton iteration.
- In this case, PreAA(m) leverages AA to accelerate Newton iteration.

using the preconditioner jac($\mathcal{F}$)



Enhancing Anderson Acceleration with Preconditioning

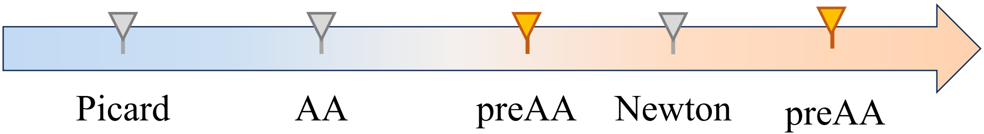
Preconditioning Strategy

We introduce a preconditioning strategy for the fixed-point iteration scheme:

$$\boldsymbol{u}_{k+1} = \boldsymbol{u}_k - \mathcal{M}_k^{-1} \mathcal{F}_k,$$

where $\mathcal{M}_k$ is a non-singular matrix, known as the **preconditioner** at iteration k.

- **Ideal preconditioner:** Close to $jac(\mathcal{F})$, yet computationally inexpensive.
- More flexibility, e.g., constant $\alpha \boldsymbol{I}$, diagonal Jacobian $diagJac(\mathcal{F})$, upper (lower) triangular Jacobian $TriU(jac(\mathcal{F}))$ and block-diagonal Jacobian $diagBlockJac(\mathcal{F})$.



Enhancing Anderson Acceleration with Preconditioning

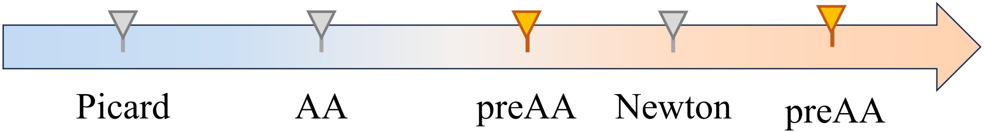
Preconditioning Strategy

We introduce a preconditioning strategy for the fixed-point iteration scheme:

$$\boldsymbol{u}_{k+1} = \boldsymbol{u}_k - \boldsymbol{\mathcal{M}}_k{}^{-1}\boldsymbol{\mathcal{F}}_k,$$

where $\boldsymbol{\mathcal{M}}_k$ is a non-singular matrix, known as the **preconditioner** at iteration k.

- **Ideal preconditioner:** Close to $jac(\boldsymbol{\mathcal{F}})$, yet computationally inexpensive.
- More flexibility, e.g., constant $\alpha\boldsymbol{I}$, diagonal Jacobian $diagJac(\boldsymbol{\mathcal{F}})$, upper (lower) triangular Jacobian $TriU(jac(\boldsymbol{\mathcal{F}}))$ and block-diagonal Jacobian $diagBlockJac(\boldsymbol{\mathcal{F}})$.
- **Delayed Update Strategy:** Reduces frequent preconditioner updates.



PreAA: Preconditioned Anderson Acceleration

Algorithm PreAA: preAA

Given $\boldsymbol{u}_0$ and window size $m \geq 1$;

Set $\boldsymbol{u}_1 = \boldsymbol{\mathcal{G}}(\boldsymbol{u}_0)$;

for $k = 1, 2, \ldots, itmax$ *until* $\|\boldsymbol{\mathcal{F}}(\mathbf{u})\| < tol$ **do**

```
// Update preconditioner
```

  if *k is evenly divisible by N_{update}* **then**

    Update preconditioning matrix $\boldsymbol{\mathcal{M}}_k$;

  Set $m_k = \min\{m, k\}$;

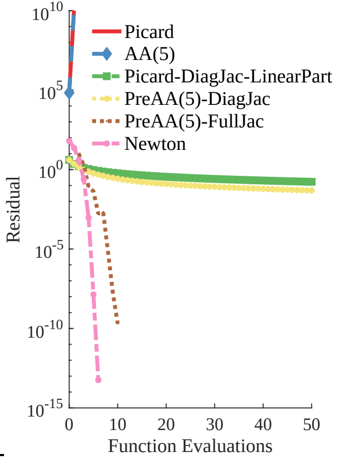
  Compute $\boldsymbol{\mathcal{E}}_k$ by solving $\boldsymbol{\mathcal{M}}_k \boldsymbol{\mathcal{E}}_k = -\boldsymbol{\mathcal{F}}_k$;

  Determine $\boldsymbol{\alpha}^{(k)} = \left(\alpha_0^{(k)}, \alpha_1^{(k)}, \ldots, \alpha_{m_k}^{(k)}\right)^{\mathrm{T}}$ that solves

$$\underset{\boldsymbol{\alpha}=(\alpha_0,\alpha_1,\ldots,\alpha_{m_k})^{\mathrm{T}}}{\arg\min} \left\| \sum_{i=0}^{m_k} \boldsymbol{\mathcal{E}}_{k-m_k+i} \boldsymbol{\alpha} \right\|_2^2, \quad \text{s.t.} \quad \sum_{i=0}^{m_k} \alpha_i = 1.$$

  Update $\boldsymbol{u}_{k+1} = \sum_{i=0}^{m_k} \alpha_i^{(k)} \left(\boldsymbol{u}_{k-m_k+i} + \boldsymbol{\mathcal{E}}_{k-m_k+i}\right)$;

Bratu problem (64×64):



Convergence Theory

Theorem (**Residual bounds**)

Assume that the non-singular preconditioner $\boldsymbol{\mathcal{M}}_k$ satisfies $\|\mathbf{I} - \boldsymbol{\mathcal{M}}_k^{-1}\boldsymbol{\mathcal{J}}_k\| \leq L_k$, then the errors generate by preconditioned Anderson acceleration algorithm satisfy

$$\|e_{k+1}\| \leq \boldsymbol{\mathcal{C}} \sum_{j=0}^{m} \|e_{k-j}\|,$$

where $\boldsymbol{\mathcal{C}} = \max\{L_k, L_{k-1}, \cdots, L_{k-m}\} \cdot \max\{2\|\Gamma_k\|_\infty, 1 + \|\Gamma_k\|_\infty\}$ with $\Gamma_k = (\gamma_1, \gamma_2, \cdots, \gamma_m)^{\mathrm{T}}$.

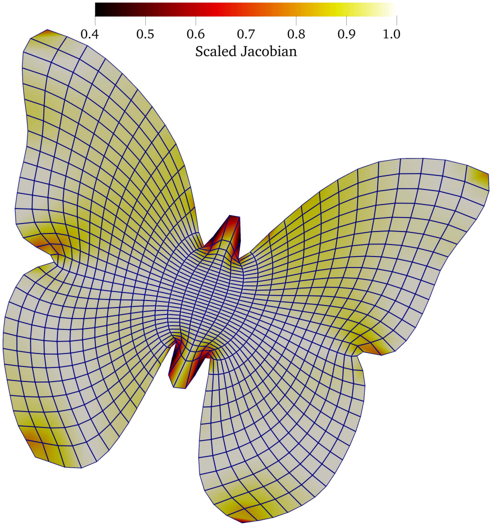
Corollary

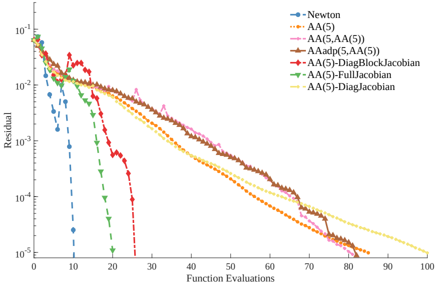
If we select a non-singular preconditioner $\boldsymbol{\mathcal{M}}_k$ that is sufficiently close to the Jacobian matrix $\boldsymbol{\mathcal{J}}_k$, such that

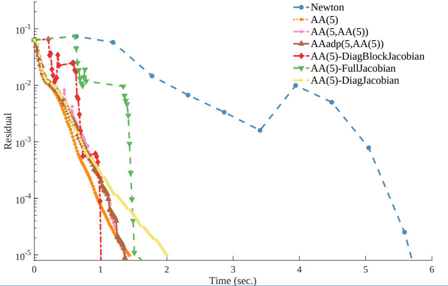
$$\left\|\mathbf{I} - \mathbf{M}_k^{-1}\boldsymbol{\mathcal{J}}_k\right\| \leq L_k < 1,$$

then the preconditioned Anderson acceleration $\mathbf{PreAA(m)}$ *converges.*

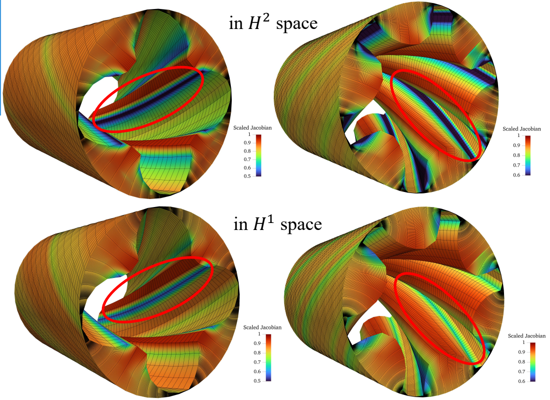
Butterfly Example: Performance Comparisons

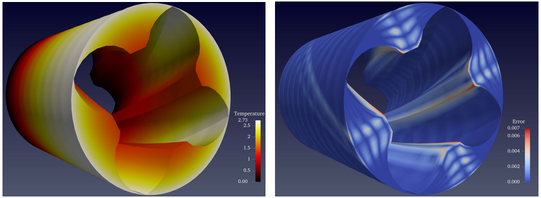




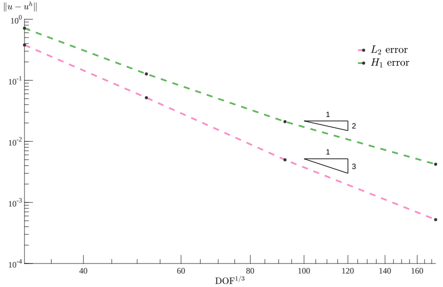


Applications to IGA simulation





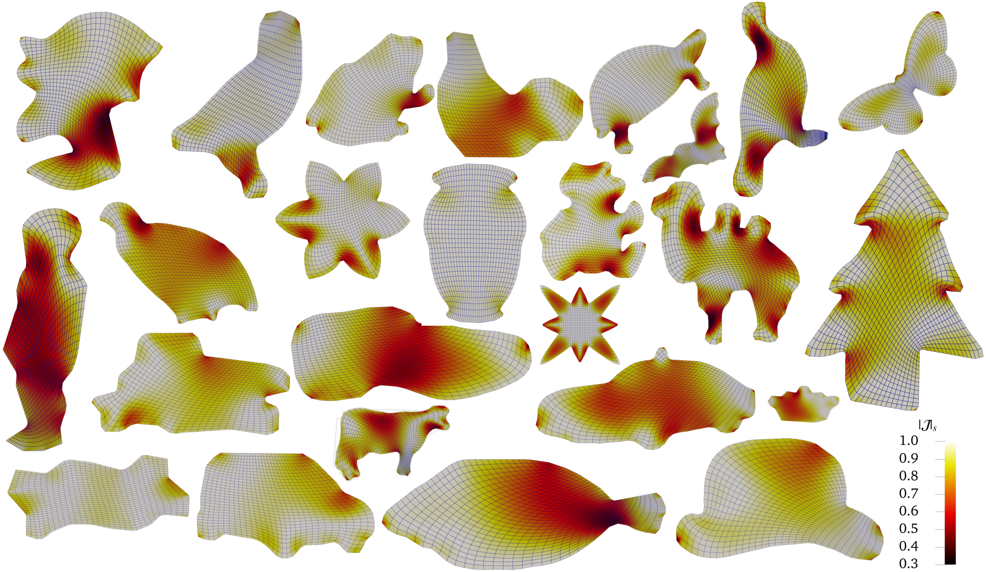
IGA solution and absolute error colormap (DOFs = 30704)



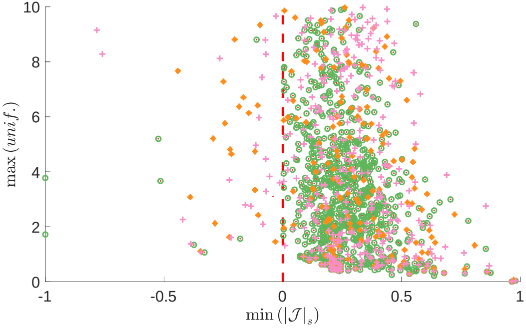
Error convergence history for h-refinement

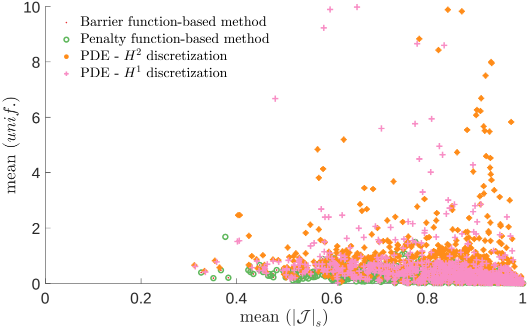
Poisson problem:
$$\begin{cases} -\Delta u = f, & \text{in } \Omega, \\ u = g, & \text{on } \Gamma_D, \\ \mathbf{n} \cdot \nabla u = h, & \text{on } \Gamma_N. \end{cases}$$

Planar Parameterization Test Dataset (977 models)



Effectiveness and Quality Assessment

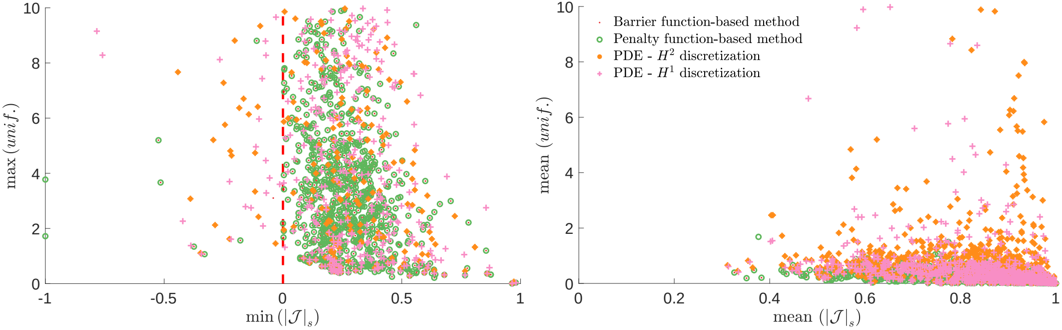




Success rates:

- PDE - H^2 discretization [Hinz+2018]: $608/977 \simeq 62.23\%$;
- PDE - H^1 discretization [Ours]: $721/977 \simeq 73.80\%$;

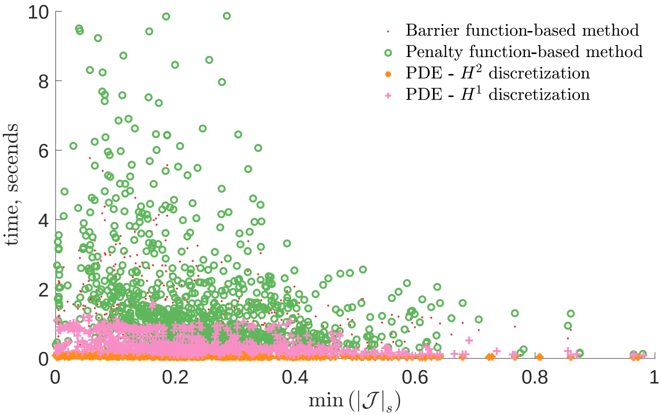
Effectiveness and Quality Assessment



Success rates:

- PDE - H^2 discretization [Hinz+2018]: $608/977 \simeq 62.23\%$;
- PDE - H^1 discretization [Ours]: $721/977 \simeq 73.80\%$;
- Barrier function-based method [Ji+2021]: $961/977 \simeq$ **98.36%**;
- Penalty function-based method [Ji+2022]: $956/977 \simeq 97.85\%$.

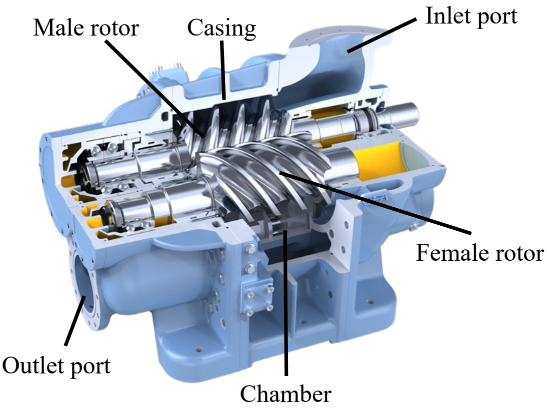
Computational Time



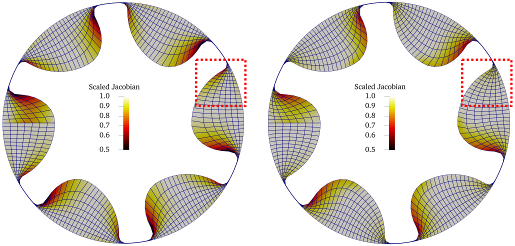
- PDE-based ∼ 0.2 sec., optimization based ∼ 2 sec. on my laptop. (DOFs=1250)
- PDE-based methods demonstrate higher efficiency.

Agenda

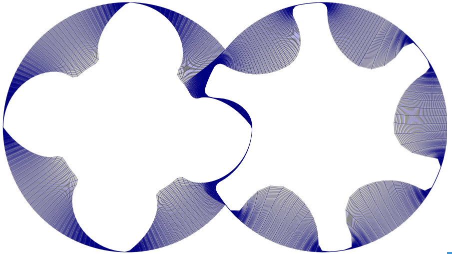
Rotary Twin-Screw Compressor Application



Rotary Twin-Screw Compressor (source [a])



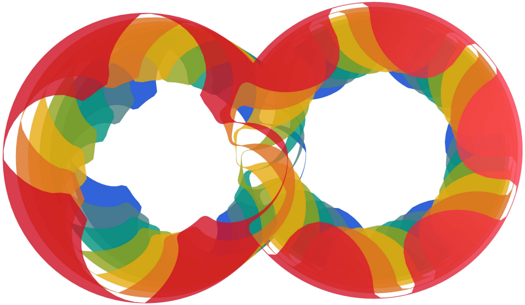
Hinz's (2020) Ours

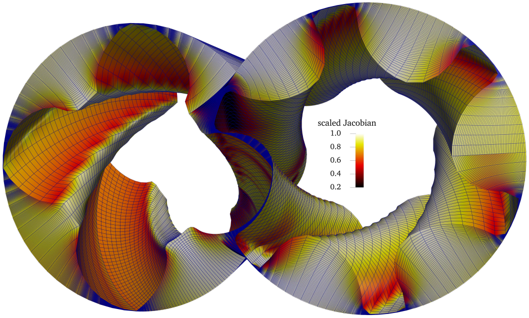


[a] https://www.gascompressors.co.uk/technologies/oil-flooded-screw-compressor/

Volumetric Completion via Spline Lofting

- Complete volumetric parameterization by lofting computed slices.
- With the **PreAA** solver, achieve parameterization for twin-screw machines **in just 3 seconds** on my personal laptop.



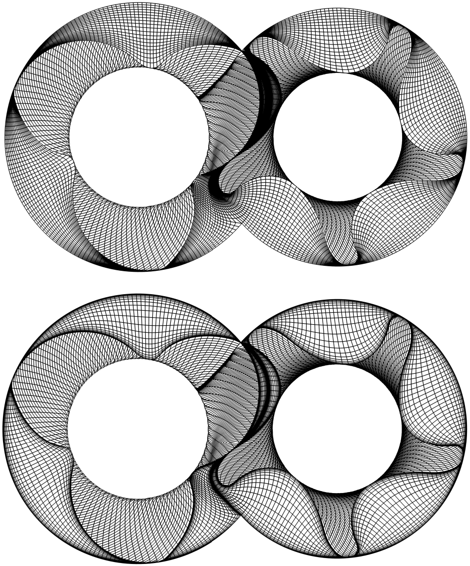


Discretization I: Boundary Layer Mesh

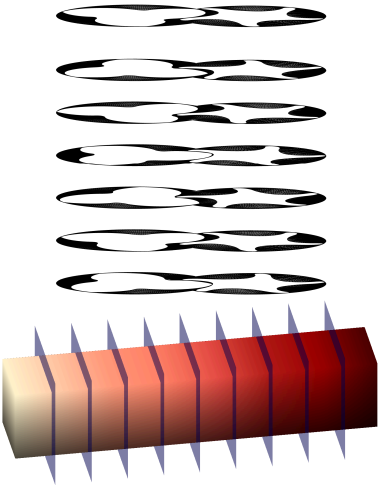
- **Down-sampling is ALL YOU NEED**.
- Employ a simple expansion transformation:

$$\begin{cases} \xi = \hat{\xi}, \\ \eta = \dfrac{\tanh\left(\alpha(2\hat{\eta}-1)\right)}{2\tanh(\hat{\eta})} + \dfrac{1}{2}, \end{cases}$$

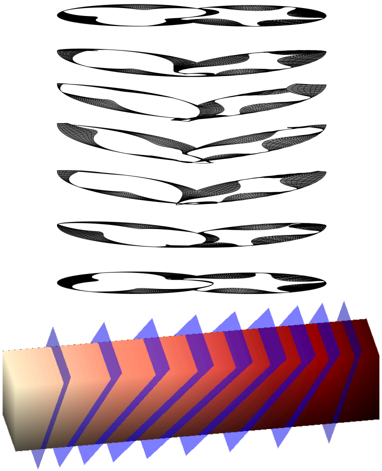
where α represents the expansion factor.



Discretization II: Flow-Aligned Hex Mesh



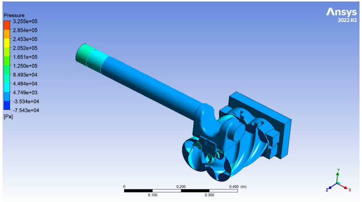
generic discretization

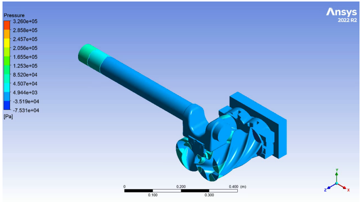


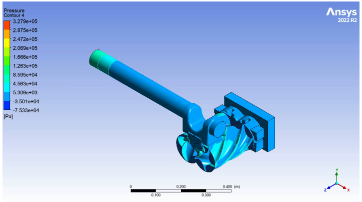
flow-aligned discretization

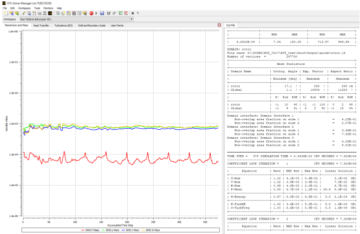
Simulation using ANASYS CFX

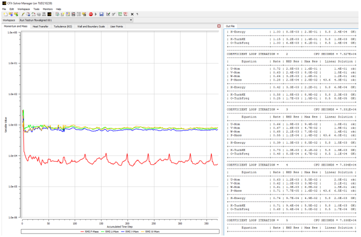
- Mesh density: $198 \times 95 \times 8$ for the male rotor and $200 \times 95 \times 8$ for the female rotor.





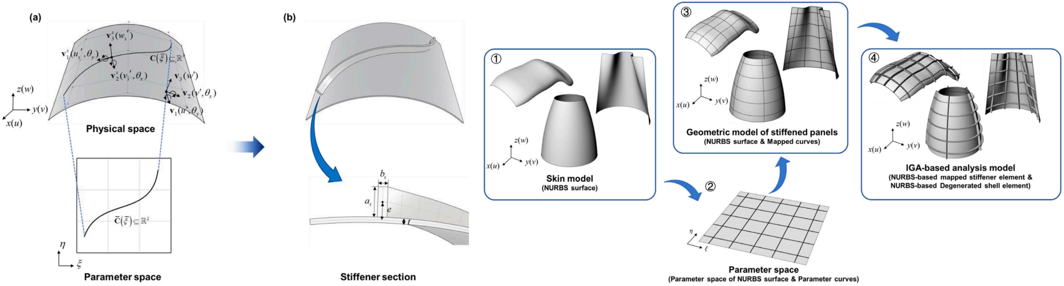






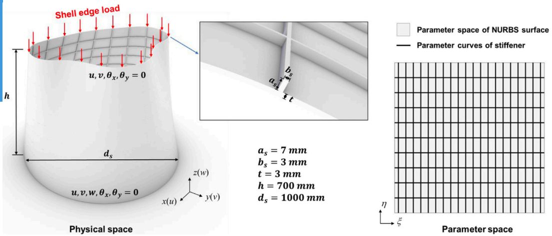
SCORG ™ Boundary Layer Mesh **Flow-aligned Mesh**

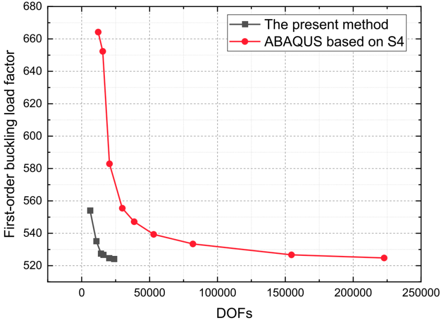
II: Isogeometric-Based Mapping Modeling for Stiffener Layout Optimization

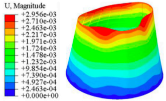
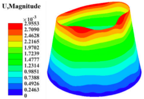
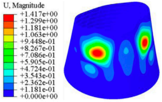
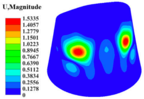
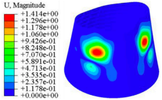
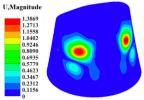


- Treat **stiffeners** not as separate entities but **as curves on the skin surface**.
- Ensures **stiffeners perfectly perpendicular to the skin** by utilizing the normal information of the NURBS surface.

Comparison with ABAQUS (S4 element)



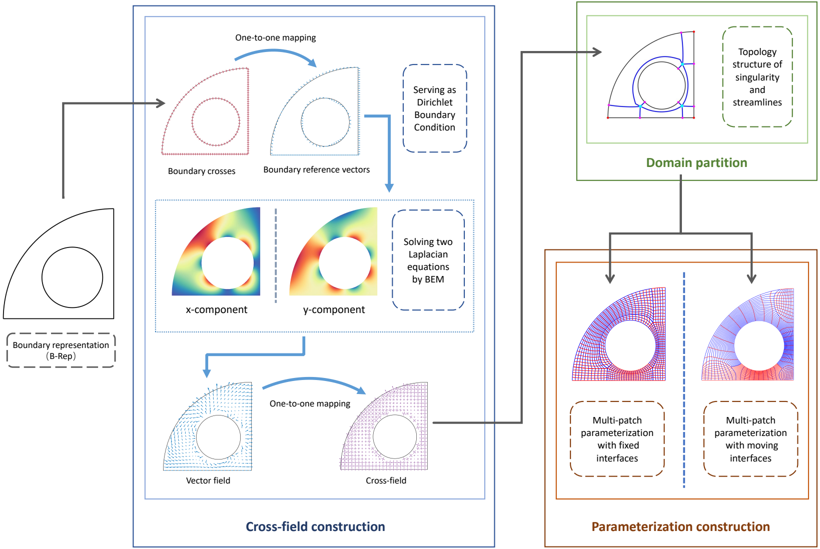


Analysis Type	Mode	FEA based on S4 by ABAQUS with 222810 DOFs (global seed size: 8 mm)	The present method by MATLAB code with 23940 DOFs
Deformation (Static Analysis)	-		
Linear Buckling Analysis	1st	 $\lambda_1 = 524.74$	 $\lambda_1 = 524.08$
	2nd	 $\lambda_2 = 535.96$	 $\lambda_2 = 534.93$

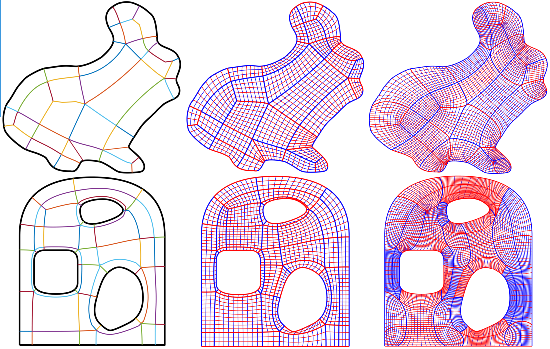
ABAQUS (DOFs=$222,810$) vs. our method (DOFs=$23,940$)

Agenda

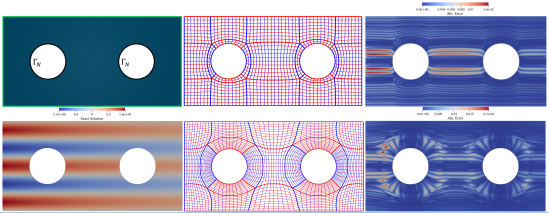
Multi-patch Parameterization using Cross-field

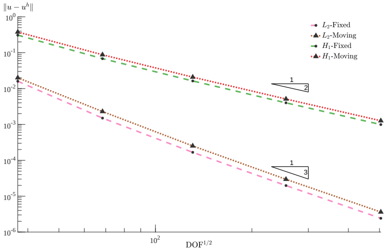


Results and Comparisons



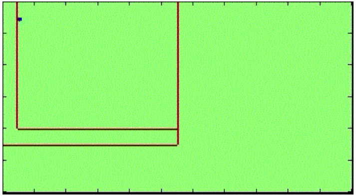
Model	#Patch	Method	$\lvert\mathcal{J}\rvert_s$ min.	$\lvert\mathcal{J}\rvert_s$ avg.	*unif.* min.	*unif.* avg.	*unif.* max.
rabbit	33	Coons	-0.8593	0.9628	0.7030	**0.9410**	1.0524
		fixed-I	**0.2204**	**0.9504**	0.6103	0.9544	**0.9982**
		moving-I	0.02918	0.9283	**0.0000**	0.9550	1.0000
3 holes	46	Coons	-0.5492	0.9710	0.8008	0.9573	1.0958
		fixed-I	**0.1545**	**0.9716**	0.8007	0.9573	0.9978
		moving-I	0.1461	0.9361	**0.6791**	**0.9571**	**0.9968**



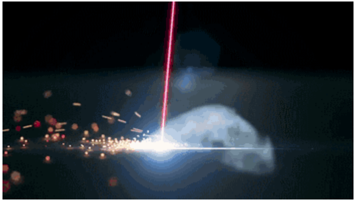


Agenda

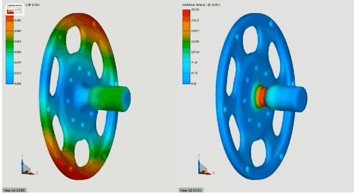
Anisotropic Phenomena in Physics



Wave Propagation. source



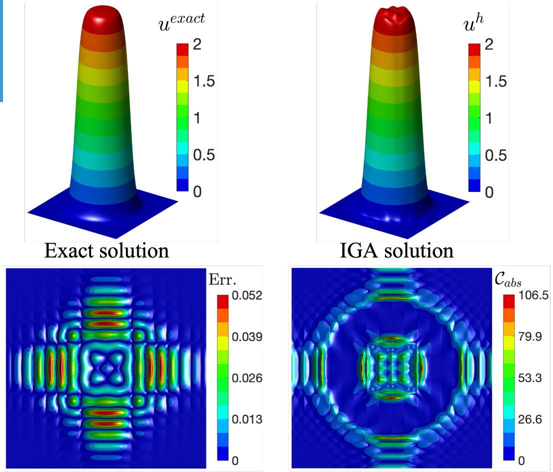
Laser Printing. source



Stress Concentration. source

- **Localized and anisotropic features extensively exist** in physical phenomena;
- Isotropic parameterizations are not efficient for such problems;
- **Anisotropic parameterizations (r-adaptivity)**
  - Enhance per-DOF accuracy while keeping constant total DOFs.

Basic Idea



Exact solution IGA solution

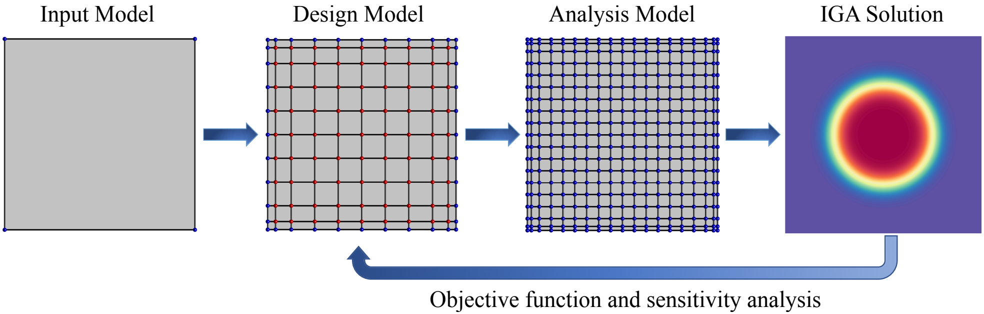
Absolute error Absolute principal curvature

- **Absolute principal curvature** captures isogeometric solution variations;
- A **tight relation** between geometric quantity and numerical solution is established;
- Absolute error and curvature show **closely aligned performance**;
- Absolute principal curvature serves as a reliable error estimator.

Basic Idea - Cont'd

Anisotropic parameterizations are often solution-dependent:

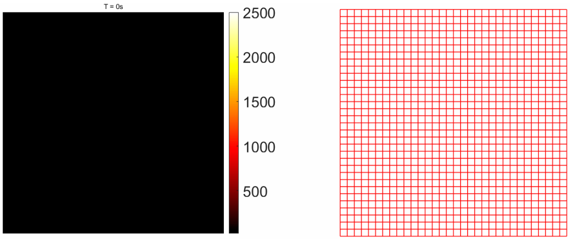
- Need **good numerical solution accuracy** to drive parameterization;
- Adjust as few control points as possible **for high efficiency**;
- **Bi-level strategy**: a coarse level (design model) to update the parameterization for efficiency's sake and a fine level (analysis model) to perform the isogeometric simulation for accuracy's sake.



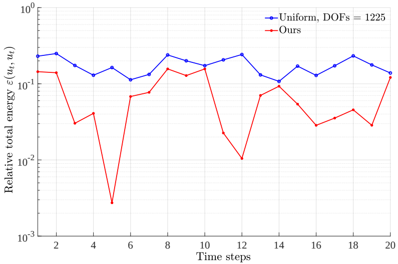
Application to Time-dependent Dynamic PDE

- Consider a 2D **linear heat transfer problem** with a moving Gaussian heat source:

$$\begin{cases} C_p \rho \dfrac{\partial u(\mathbf{x},t)}{\partial t} - \nabla \cdot (\kappa \nabla u(\boldsymbol{u}, t)) = f(\mathbf{x}, t) & \text{in } \Omega \times T \\ u(\mathbf{x}, t) = u_0 & \text{in } \Omega \\ \kappa \nabla u(\mathbf{x}, t) = 0 & \text{on } \partial\Omega \times T \end{cases}$$



$u(\mathbf{x}, t)$ and the parameterizations



Errors vs. time instants t

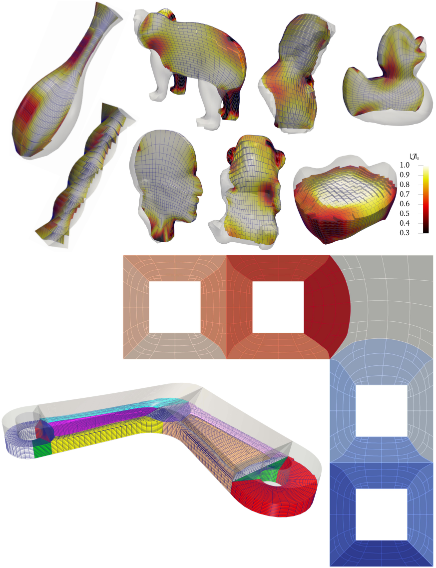
Agenda

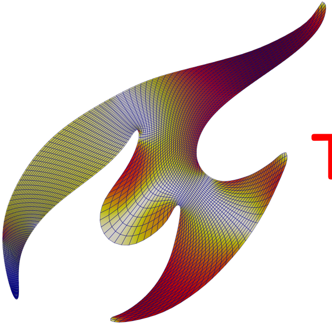
Conclusions and Outlook

- Proposed sereval **efficient and robust isotropic approaches**;
- Proposed a **curvature-based r-adaptive method using bi-level approach**;
- Provided an **open-source implementation** within the G+Smo platform.

Future Work:

- **Topology Computation:** Investigate multi-patch parameterization.
- **Adaptive Methods:** Enhance efficiency.





Thanks for Your Attention!

Q&A.

If interested in my research, please feel free to contact me! ;-)

- Email: jiyess@outlook.com
- Homepage: https://jiyess.github.io
- GitHub: jiyess