



On an improved PDE-based parameterization method for IsoGeometric Analysis (IGA) using preconditioned Anderson acceleration

Delft University of Technology

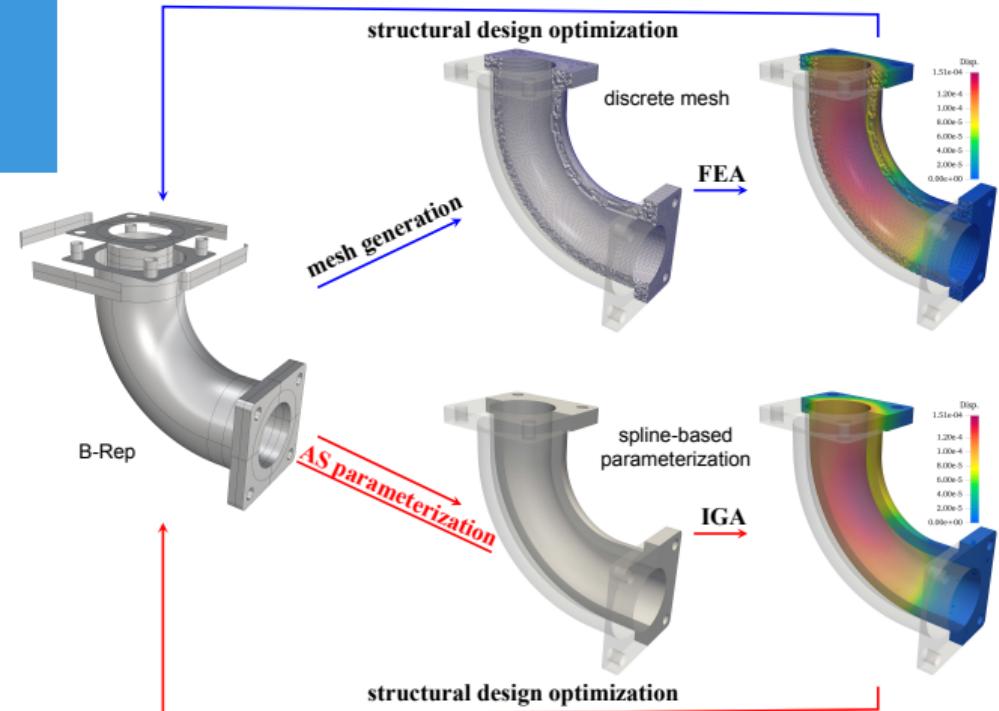
Ye Ji, Kewang Chen, Matthias Möller, Cornelis Vuik

5–7 July 2023, Genoa, Italy, GMP 2023

Agenda

- ① Research Background and Motivation
- ② Parameterization Quality Improvement: Scaled H^1 Discretization
- ③ Enhancing Computational Efficiency: Preconditioned Anderson Acceleration
- ④ Results and Applications
- ⑤ Conclusions and Outlook

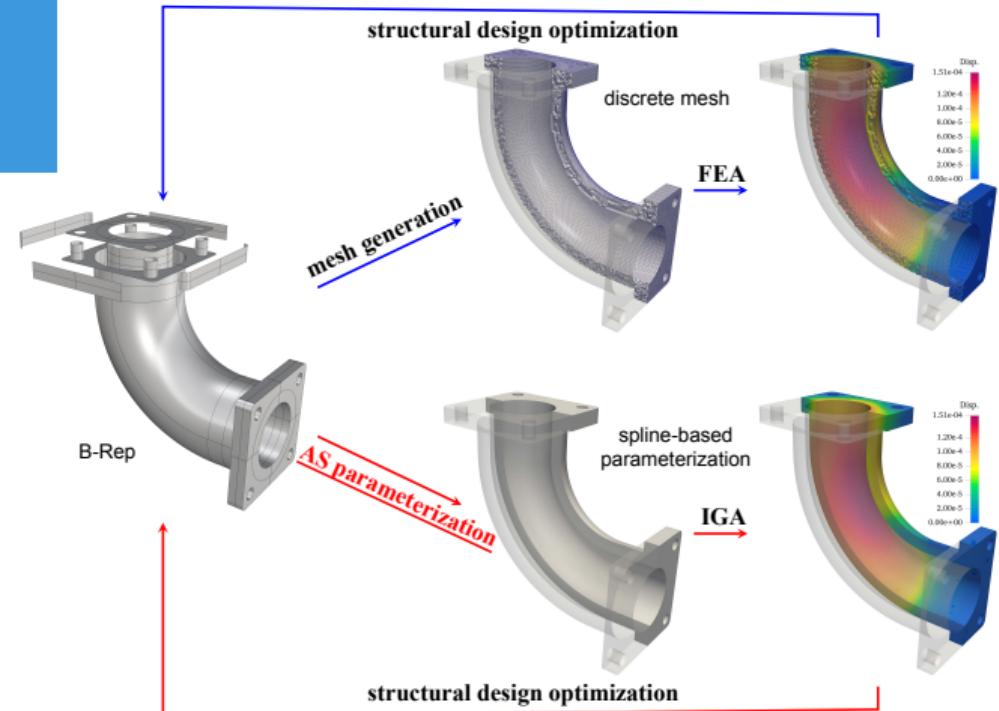
IsoGeometric Analysis (IGA)



- Introduced by T.J.R. Hughes et al., 2005.
- **Key Idea:** Approximate physical fields with the **same basis functions** used in CAD model representation.

Design-analysis-optimization product development workflow

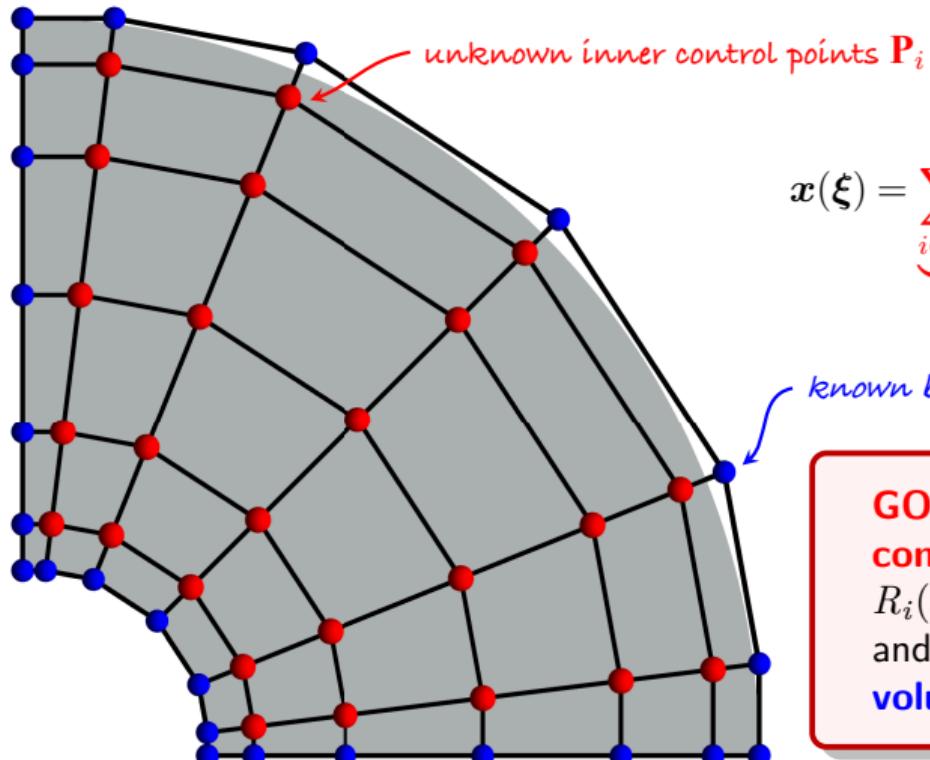
IsoGeometric Analysis (IGA)



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



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

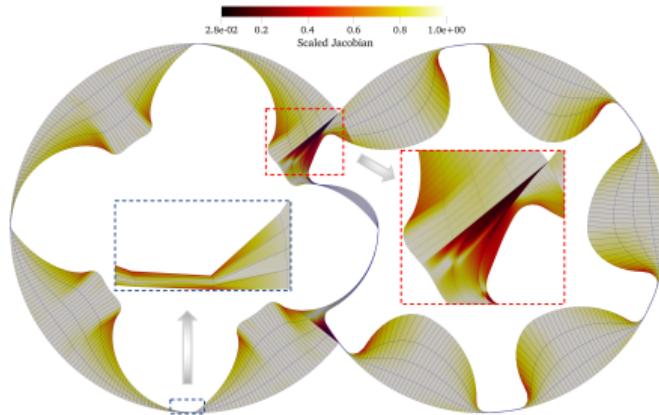
known boundary control points \mathbf{P}_j

GOAL: Construct the **unknown inner control points \mathbf{P}_i** (or basis functions $R_i(\xi)$) such that x ensures **bijection** and exhibits **minimal angle and area-/volume distortion**.

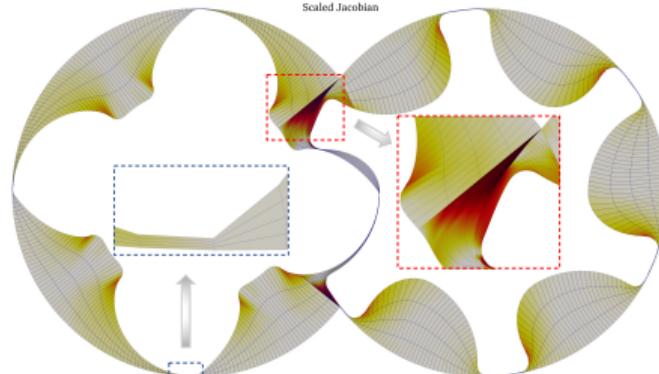
Rotary twin-screw machines

Rotary twin-screw compressor [source](#)

Rotary twin-screw machines



Penalty function-based method ^a



Elliptic Grid Generation method ^b

Rotary twin-screw compressor [source](#)

^aRef.: Ji, Y. et al. (2022). Penalty function-based volumetric... Computer Aided Geometric Design, 94, 102081.

^bRef.: Hinz, J.P. et al. (2018). Elliptic grid generation techniques... Computer Aided Geometric Design, 65, 48-75.

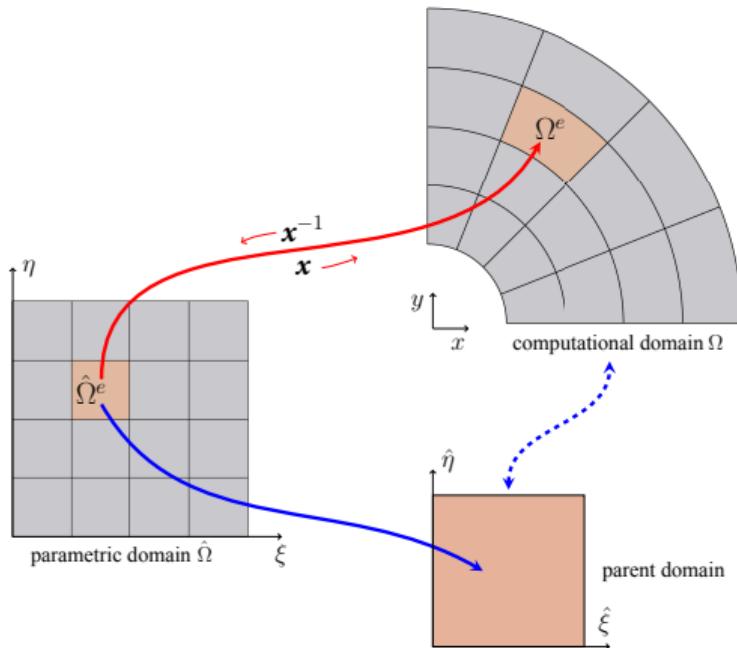
Agenda

- ① Research Background and Motivation
- ② Parameterization Quality Improvement: Scaled H^1 Discretization
- ③ Enhancing Computational Efficiency: Preconditioned Anderson Acceleration
- ④ Results and Applications
- ⑤ Conclusions and Outlook

Elliptic Grid Generation (EGG) method

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

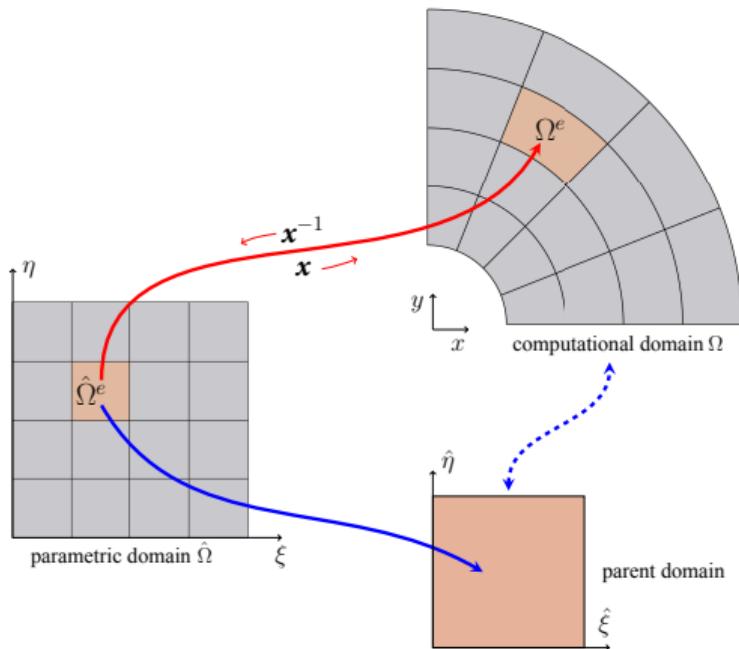
$$\begin{cases} \nabla^2 \xi(x, y) = 0 \\ \nabla^2 \eta(x, y) = 0 \end{cases} \quad \text{s.t. } \mathbf{x}^{-1}|_{\partial\Omega} = \partial\hat{\Omega}.$$



Elliptic Grid Generation (EGG) method

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

$$\begin{cases} \nabla^2 \xi(x, y) = 0 \\ \nabla^2 \eta(x, y) = 0 \end{cases} \quad \text{s.t. } \mathbf{x}^{-1}|_{\partial\Omega} = \partial\hat{\Omega}.$$

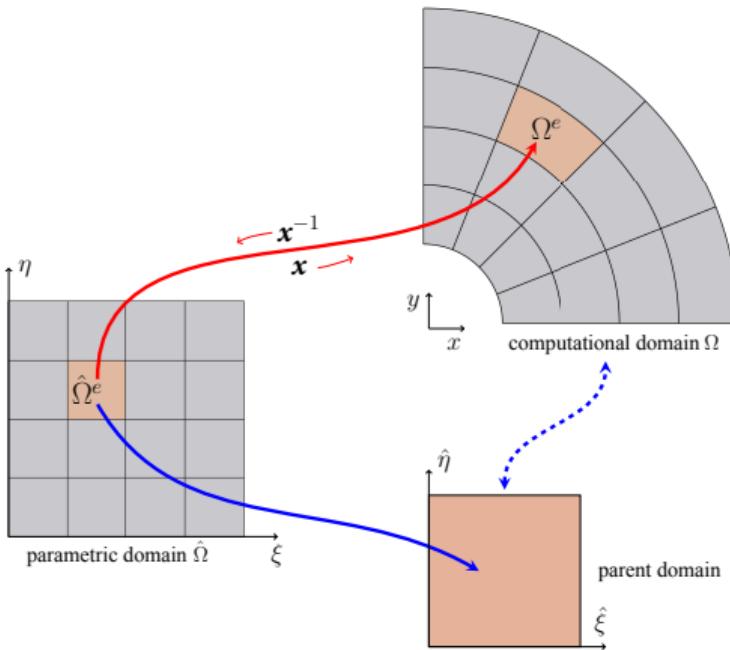


- The **existence and uniqueness** of the harmonic mapping \mathbf{x}^{-1} is guaranteed if ^a:
 - The curvature of $\hat{\Omega}$ is non-positive;
 - The boundary $\hat{\Omega}$, when considered with respect to the metric on Ω , is convex.

Elliptic Grid Generation (EGG) method

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

$$\begin{cases} \nabla^2 \xi(x, y) = 0 \\ \nabla^2 \eta(x, y) = 0 \end{cases} \quad \text{s.t. } x^{-1}|_{\partial\Omega} = \partial\hat{\Omega}.$$



- The **existence and uniqueness** of the harmonic mapping 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 x^{-1} offers a **one-to-one mapping** (with the Jacobian J not vanishing), which is ensured by the Radó-Kneser-Choquet theorem. ^b

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

^b Ref.: 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 $\mathbf{x} : \hat{\Omega} \rightarrow \Omega$ by solving Laplace equations:

$$\begin{cases} \nabla^2 \xi(x, y) = 0 \\ \nabla^2 \eta(x, y) = 0 \end{cases} \quad \text{s.t. } \mathbf{x}^{-1}|_{\partial\Omega} = \partial\hat{\Omega}.$$

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

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

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

$$\begin{cases} \nabla^2 \xi(x, y) = 0 \\ \nabla^2 \eta(x, y) = 0 \end{cases} \quad \text{s.t. } \mathbf{x}^{-1}|_{\partial\Omega} = \partial\hat{\Omega}.$$

- Nonlinear vector-valued second-order PDE ¹

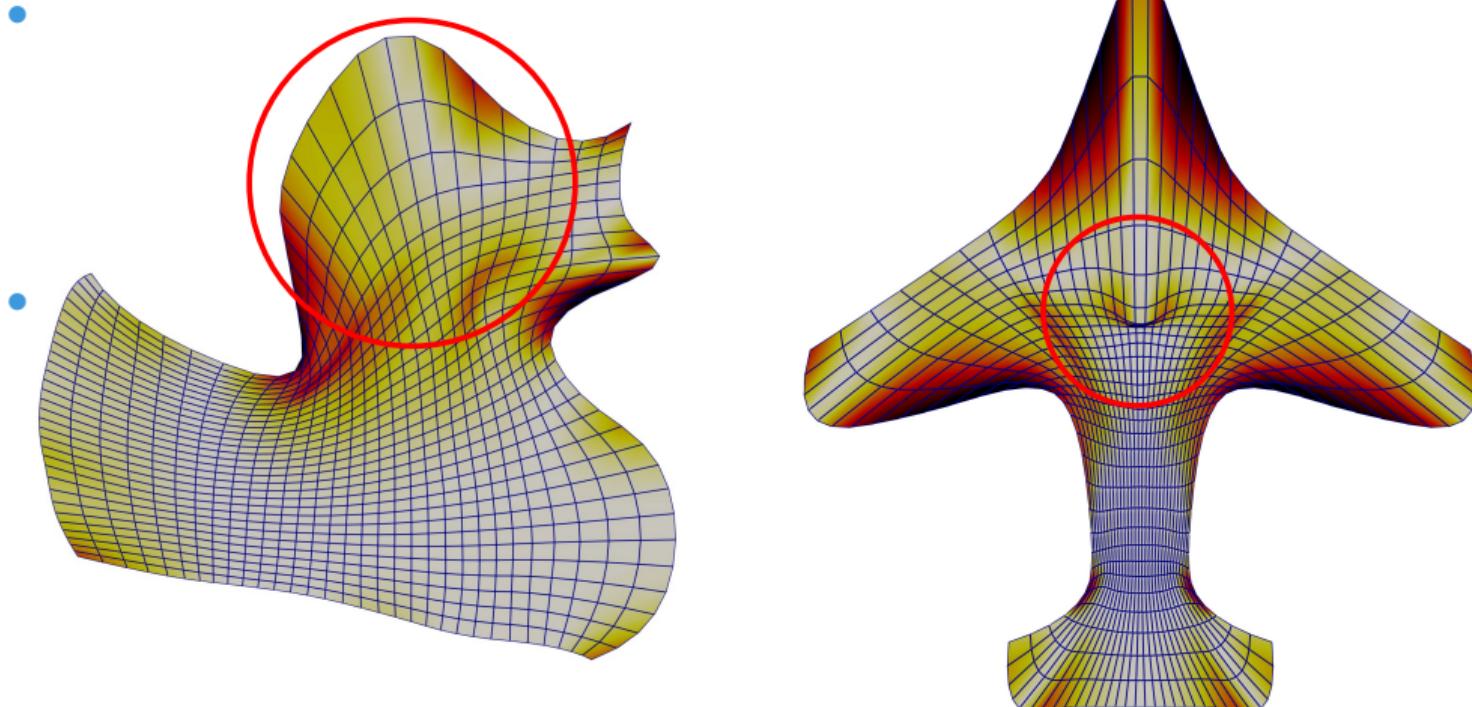
$$\forall R_i \in \Sigma_0 : \begin{cases} \int_{\hat{\Omega}} \mathbf{R} \tilde{\mathcal{L}} \mathbf{x} \, d\hat{\Omega} = \mathbf{0}, \\ \int_{\hat{\Omega}} \mathbf{R} \tilde{\mathcal{L}} \mathbf{y} \, d\hat{\Omega} = \mathbf{0}, \end{cases} \quad \text{s.t. } \mathbf{x}|_{\partial\hat{\Omega}} = \partial\Omega,$$

where

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

¹Hinz, 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.

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

Discretization in Sobolev space H^1

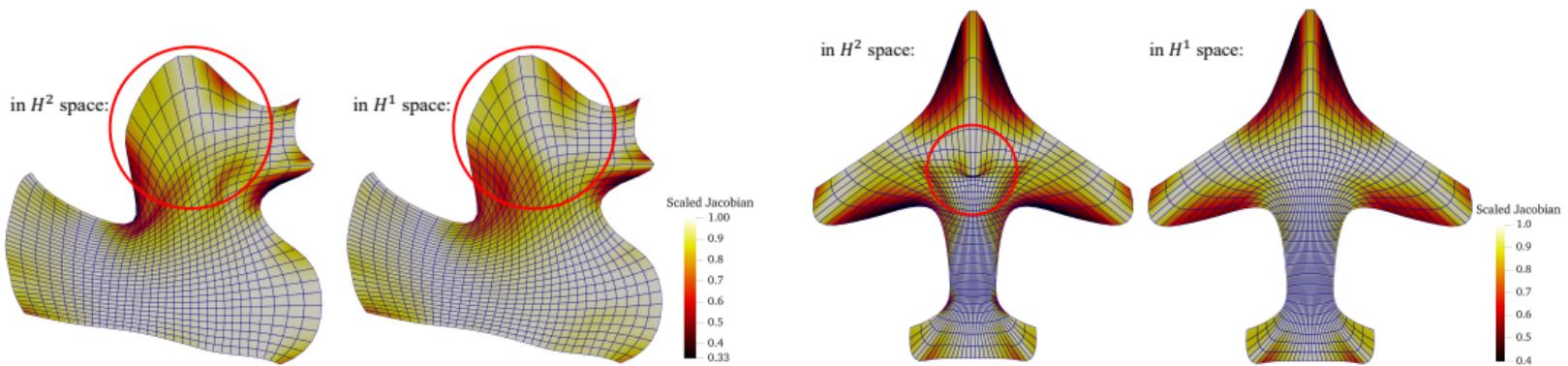
- The variational formulation in the Sobolev space H^1 reads

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

where

$$\mathbf{F}_{H^1}^x = \int_{\hat{\Omega}} \nabla_{\mathbf{x}} \mathbf{R} \cdot \nabla_{\mathbf{x}} \xi |\mathcal{J}| d\hat{\Omega}, \quad \mathbf{F}_{H^1}^y = \int_{\hat{\Omega}} \nabla_{\mathbf{x}} \mathbf{R} \cdot \nabla_{\mathbf{x}} \eta |\mathcal{J}| d\hat{\Omega},$$

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



Discretization in Sobolev space H^1

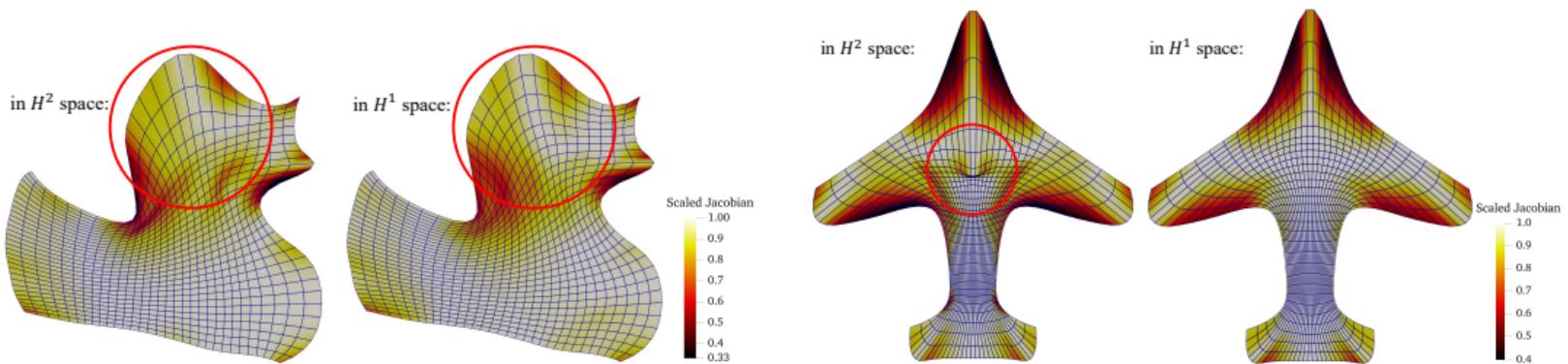
- The variational formulation in the Sobolev space H^1 reads

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

where

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

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



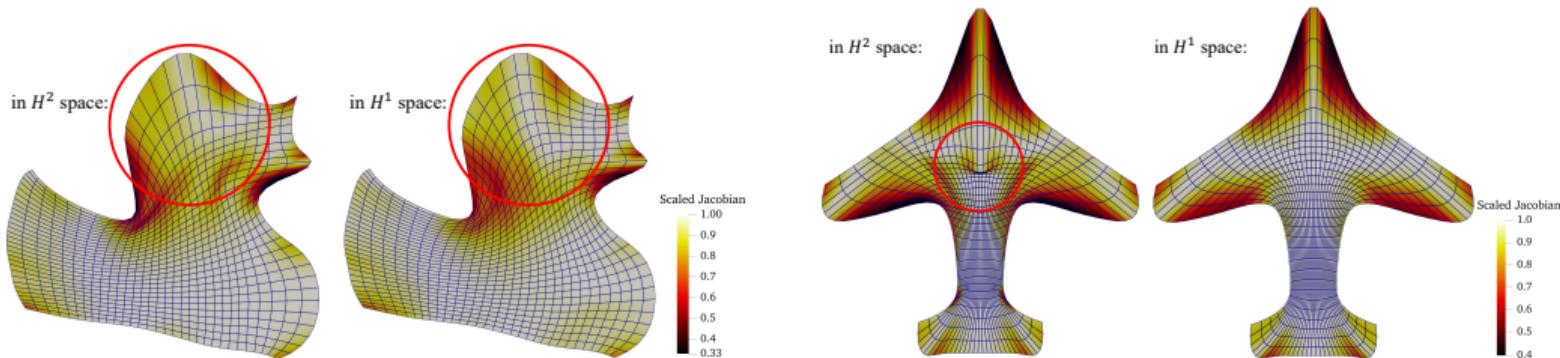
Why it Works?

- Essentially, we incorporate a monitor function M into our governing equation ¹:

$$-\nabla \cdot (M \nabla \xi) = 0.$$

where the monitor function is defined as

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



¹Gained valuable insights from a discussion with Jochen Peter Hinz earlier this week.

Agenda

- ① Research Background and Motivation
- ② Parameterization Quality Improvement: Scaled H^1 Discretization
- ③ Enhancing Computational Efficiency: Preconditioned Anderson Acceleration
- ④ Results and Applications
- ⑤ Conclusions and Outlook

Picard Iteration

Nonlinear system \iff Fixed-point iteration :
 $\mathcal{F}(\mathbf{u}) = 0 \iff \mathbf{u} = \mathcal{G}(\mathbf{u}) = \mathbf{u} + \mathcal{F}(\mathbf{u})$
for some $\mathcal{F} : \mathbb{R}^n \rightarrow \mathbb{R}^n$ and $\mathcal{G} : \mathbb{R}^n \rightarrow \mathbb{R}^n$.

- Basic Fixed-Point Iteration:

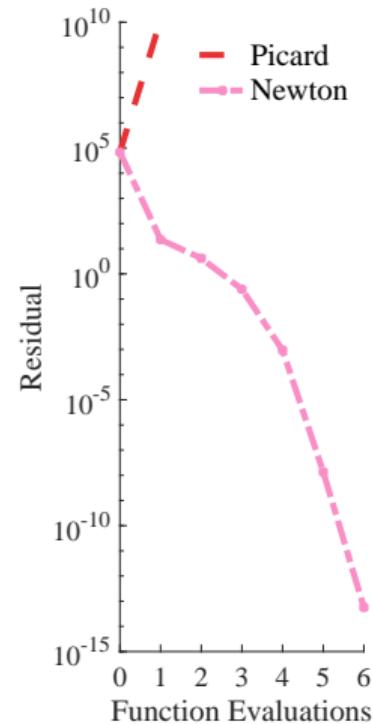
Algorithm Picard: Picard Iteration

- 1 Given \mathbf{u}_0 .
 - 2 **for** $k = 1, 2, \dots, itmax$ until $\|\mathcal{F}_k\| < tol$ **do**
 - 3 └ Set $\mathbf{u}_{k+1} = \mathcal{G}(\mathbf{u}^k)$
-

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

Bratu problem (64×64):
 $\Delta u + \lambda e^u = 0, \text{ in } \Omega,$

$$u = 0, \text{ on } \partial\Omega.$$



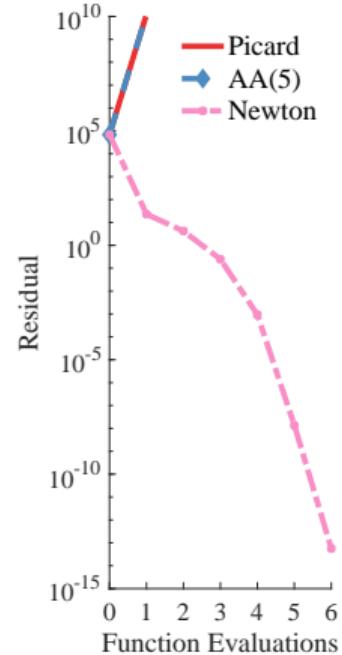
Anderson Acceleration (D.G. Anderson, 1965)

- 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

- 1 Given \mathbf{u}_0 and window size $m \geq 1$;
- 2 Set $\mathbf{u}_1 = \mathcal{G}(\mathbf{u}_0)$;
- 3 **for** $k = 1, 2, \dots, itmax$ until $\|\mathcal{F}_k\| < tol$ **do**
- 4 Set $m_k = \min\{m, k\}$;
- 5 Determine $\boldsymbol{\alpha}^{(k)} = (\alpha_0^{(k)}, \alpha_1^{(k)}, \dots, \alpha_{m_k}^{(k)})^T$ that solves
- 6
$$\arg \min_{\boldsymbol{\alpha}=(\alpha_0, \alpha_1, \dots, \alpha_{m_k})^T} \left\| \sum_{i=0}^{m_k} \mathcal{F}_{k-m_k+i} \boldsymbol{\alpha} \right\|_2^2, \quad \text{s.t.} \quad \sum_{i=0}^{m_k} \alpha_i = 1;$$
- 7 Update $\mathbf{u}_{k+1} = \sum_{i=0}^{m_k} \alpha_i^{(k)} \mathcal{G}_{k-m_k+i}$;

Bratu problem (64×64):



Enhancing Anderson Acceleration with Preconditioning

Preconditioning Strategy

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

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

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

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

using the preconditioner I



Enhancing Anderson Acceleration with Preconditioning

Preconditioning Strategy

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

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

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

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

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



Enhancing Anderson Acceleration with Preconditioning

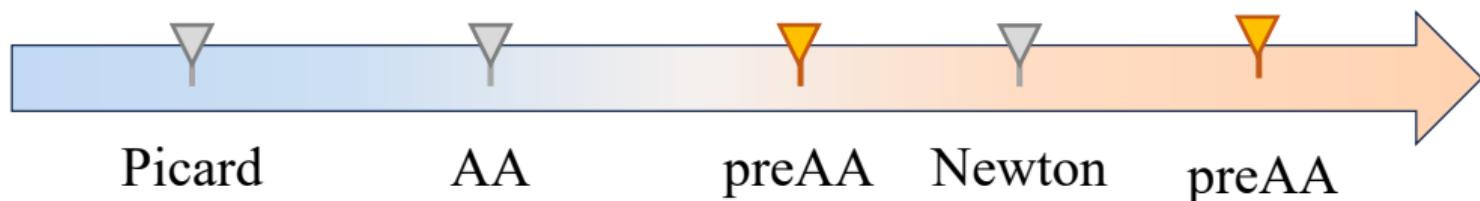
Preconditioning Strategy

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

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

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

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



Enhancing Anderson Acceleration with Preconditioning

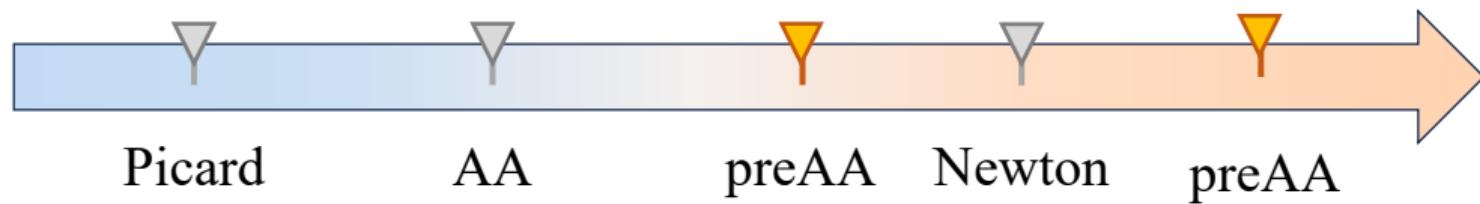
Preconditioning Strategy

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

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

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

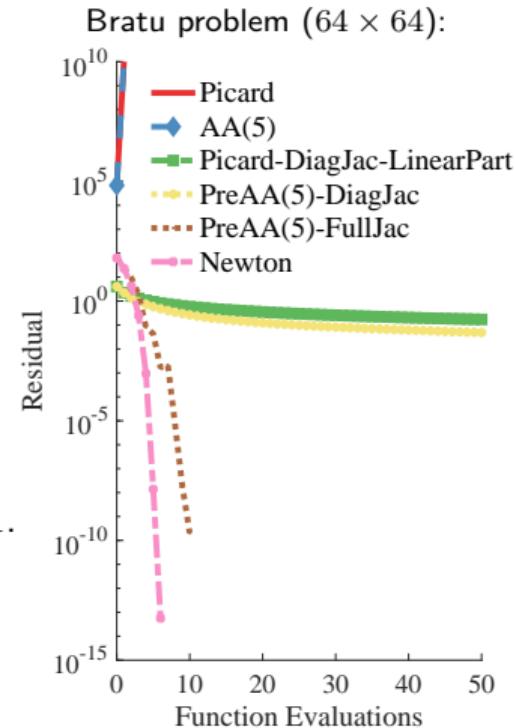
- **Ideal preconditioner:** Close to $\text{jac}(\mathcal{F})$, yet computationally inexpensive.
- More flexibility, e.g., constant $\alpha\mathbf{I}$, diagonal Jacobian $\text{diagJac}(\mathcal{F})$, upper (lower) triangular Jacobian $\text{TriU}(\text{jac}(\mathcal{F}))$ and block-diagonal Jacobian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



PreAA: Preconditioned Anderson Acceleration

Algorithm PreAA: preAA

```
1 Given  $\mathbf{u}_0$  and window size  $m \geq 1$ ;  
2 Set  $\mathbf{u}_1 = \mathcal{G}(\mathbf{u}_0)$ ;  
3 for  $k = 1, 2, \dots, itmax$  until  $\|\mathcal{F}(\mathbf{u})\| < tol$  do  
    // Update preconditioner  
    4 if  $k$  is evenly divisible by  $N_{update}$  then  
        5     Update preconditioning matrix  $\mathcal{M}_k$ ;  
    6 Set  $m_k = \min\{m, k\}$ ;  
    7 Compute  $\mathcal{E}_k$  by solving  $\mathcal{M}_k \mathcal{E}_k = -\mathcal{F}_k$ ;  
    8 Determine  $\boldsymbol{\alpha}^{(k)} = (\alpha_0^{(k)}, \alpha_1^{(k)}, \dots, \alpha_{m_k}^{(k)})^T$  that solves  
        9         
$$\arg \min_{\boldsymbol{\alpha}=(\alpha_0, \alpha_1, \dots, \alpha_{m_k})^T} \left\| \sum_{i=0}^{m_k} \mathcal{E}_{k-m_k+i} \boldsymbol{\alpha} \right\|_2^2, \quad \text{s.t.} \quad \sum_{i=0}^{m_k} \alpha_i = 1.$$
  
    10 Update  $\mathbf{u}_{k+1} = \sum_{i=0}^{m_k} \alpha_i^{(k)} (\mathbf{u}_{k-m_k+i} + \mathcal{E}_{k-m_k+i})$ ;
```



Convergence Theory

Theorem (Residual bounds)

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

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

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

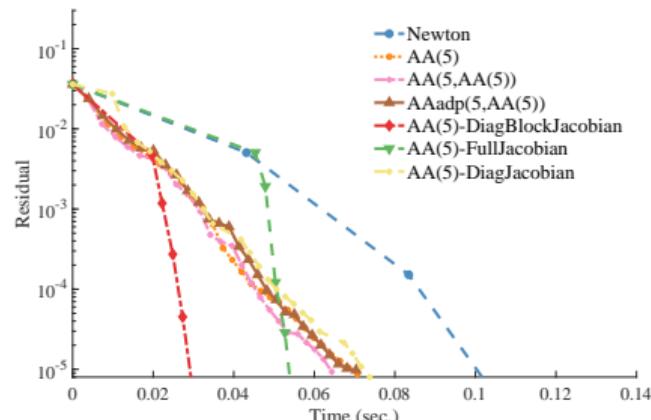
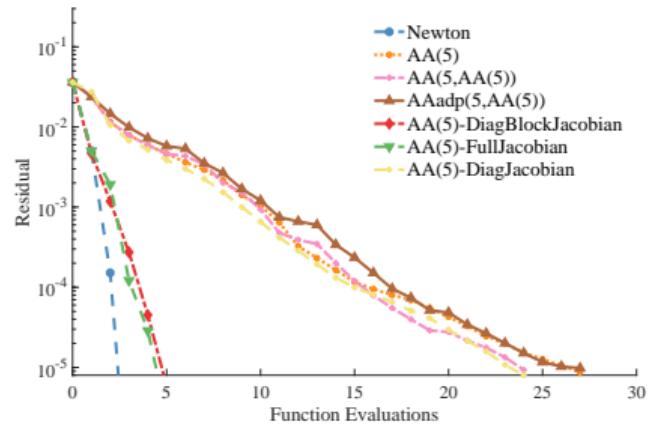
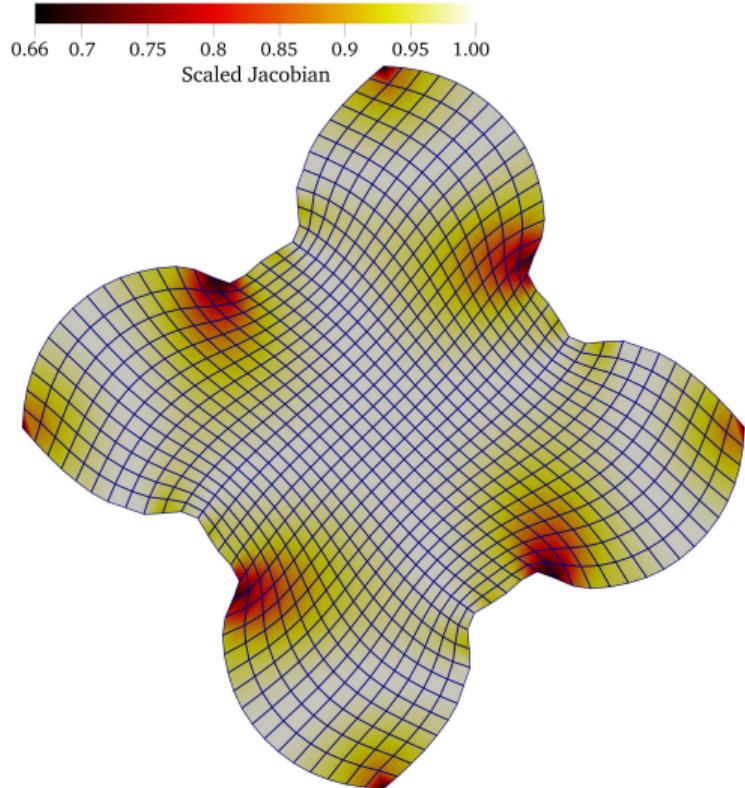
Corollary

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

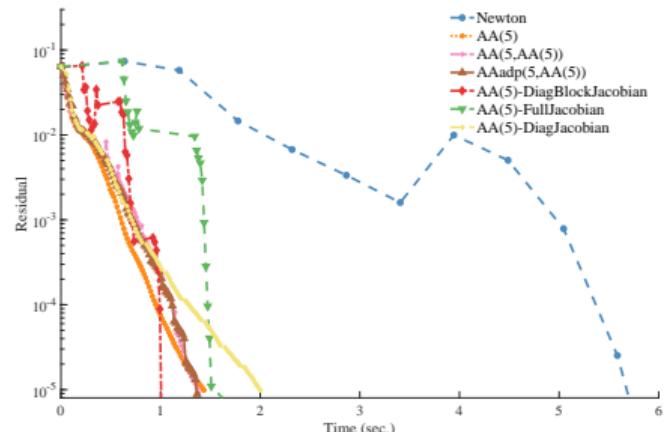
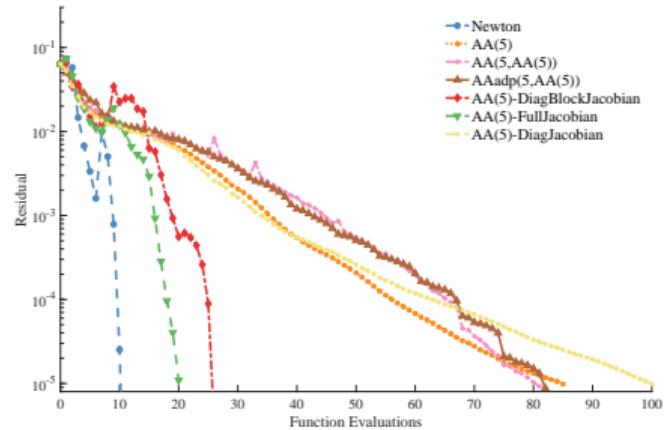
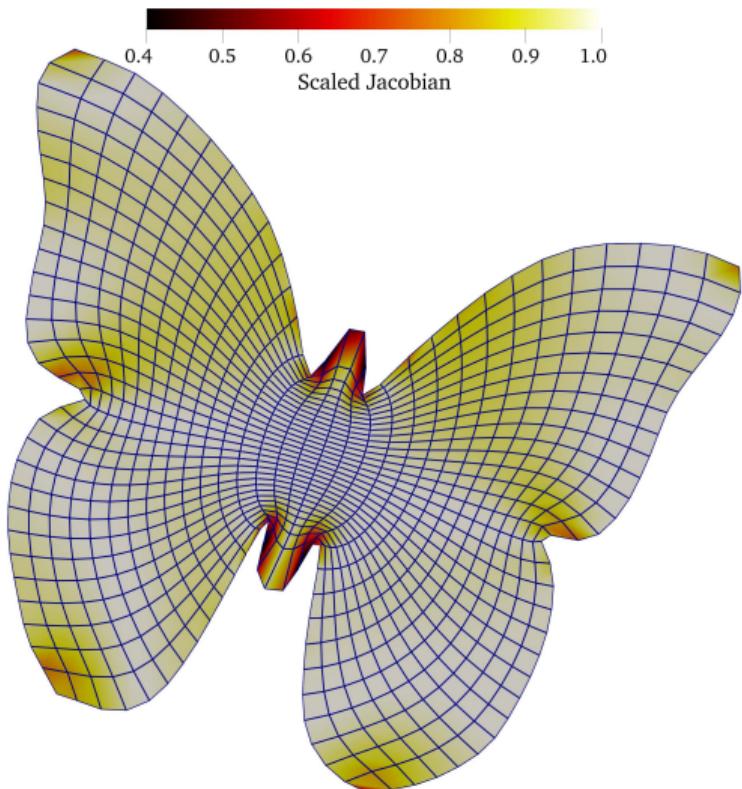
$$\|\mathbf{I} - \mathcal{M}_k^{-1} \mathcal{J}_k\| \leq L_k < 1,$$

then the preconditioned Anderson acceleration **PreAA(m)** converges.

Male rotor example: performance comparisons



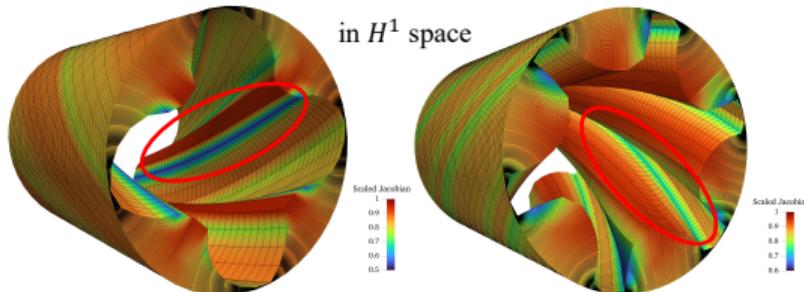
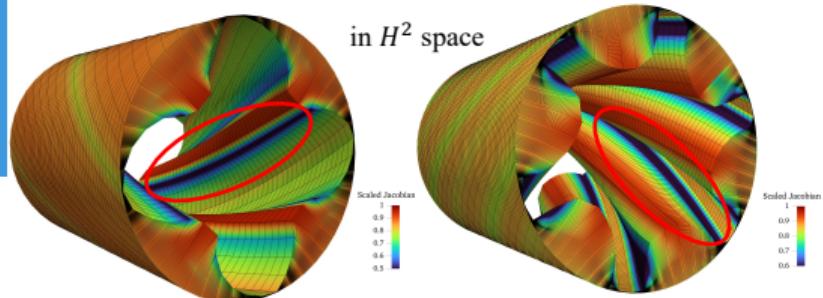
Butterfly Example: Performance Comparisons



Agenda

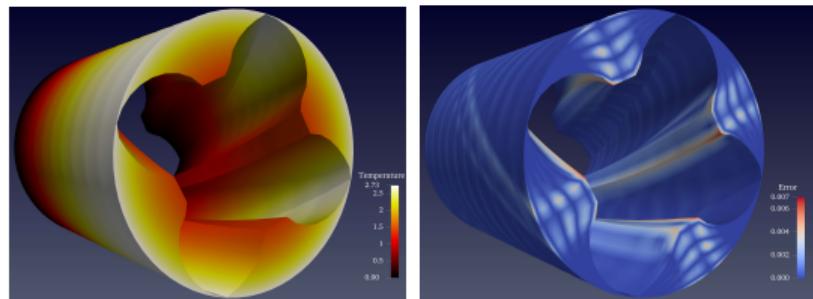
- ① Research Background and Motivation
- ② Parameterization Quality Improvement: Scaled H^1 Discretization
- ③ Enhancing Computational Efficiency: Preconditioned Anderson Acceleration
- ④ Results and Applications
- ⑤ Conclusions and Outlook

Applications to IGA simulation

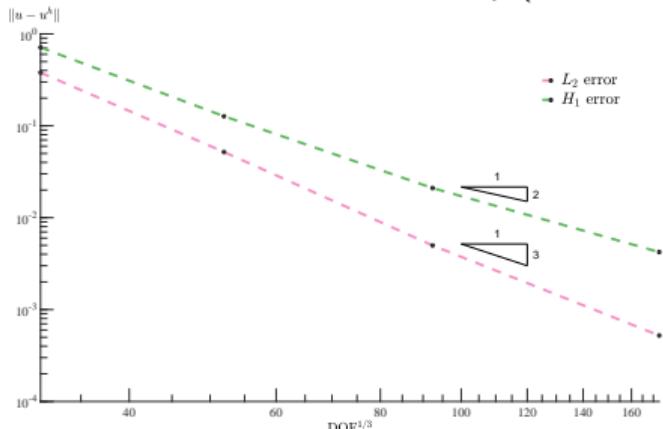


Poisson problem:

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



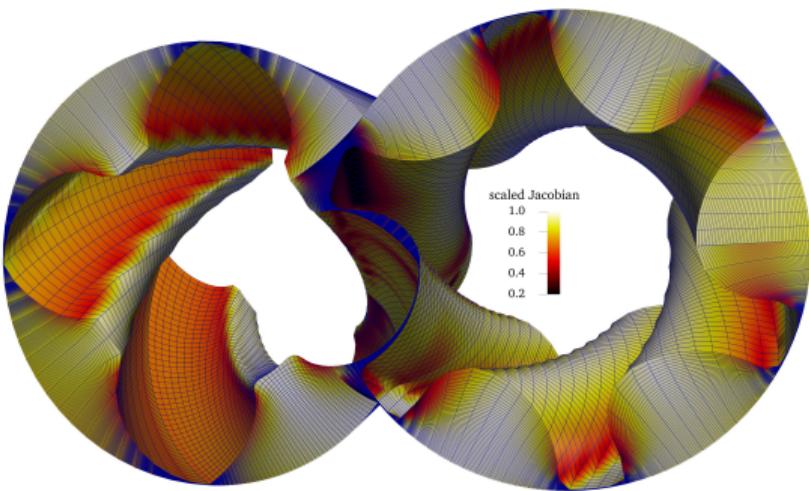
IGA solution and absolute error colormap (DOFs = 30704)



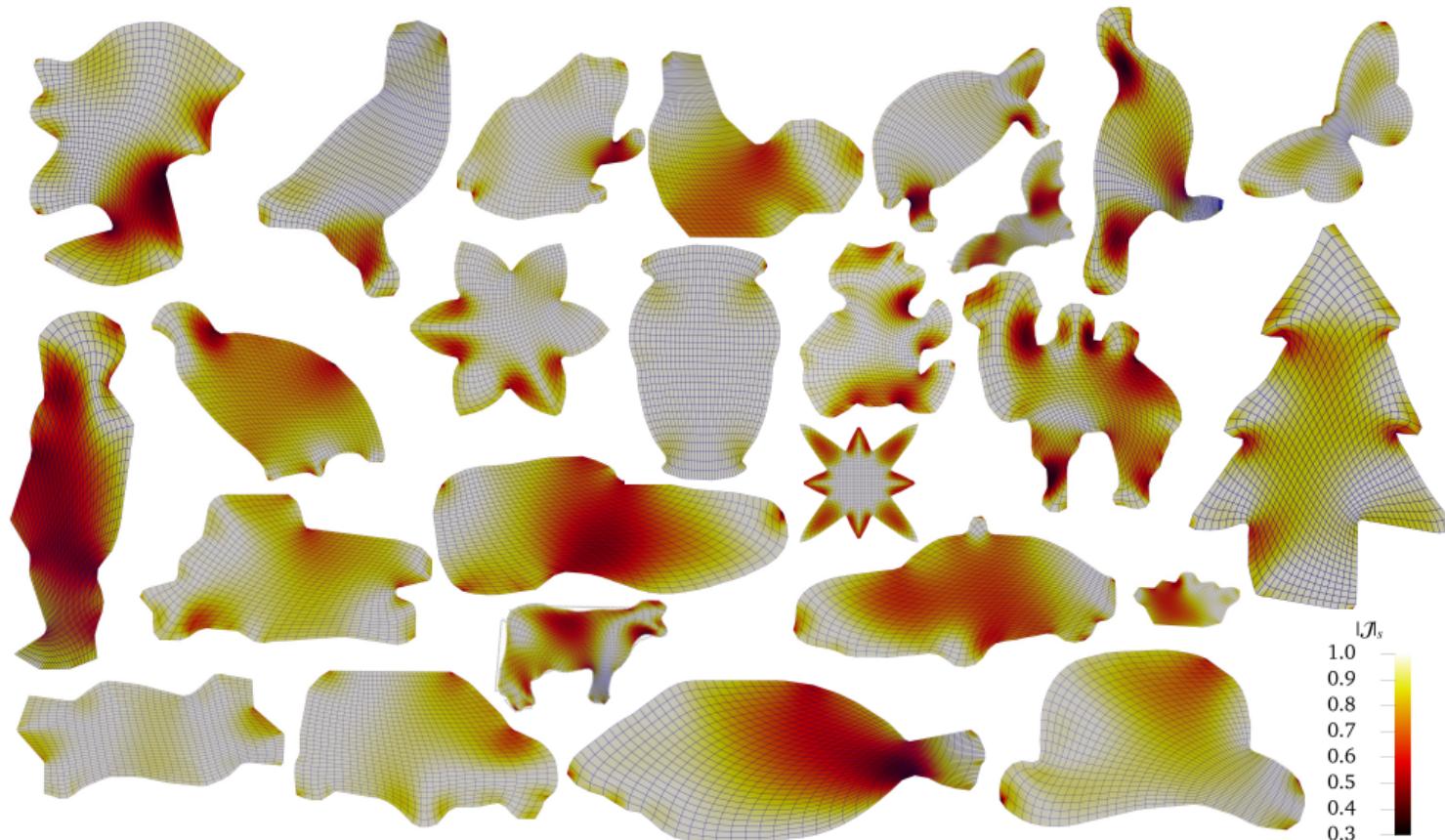
Error convergence history for h-refinement

Rotary Twin-Screw Compressor Application

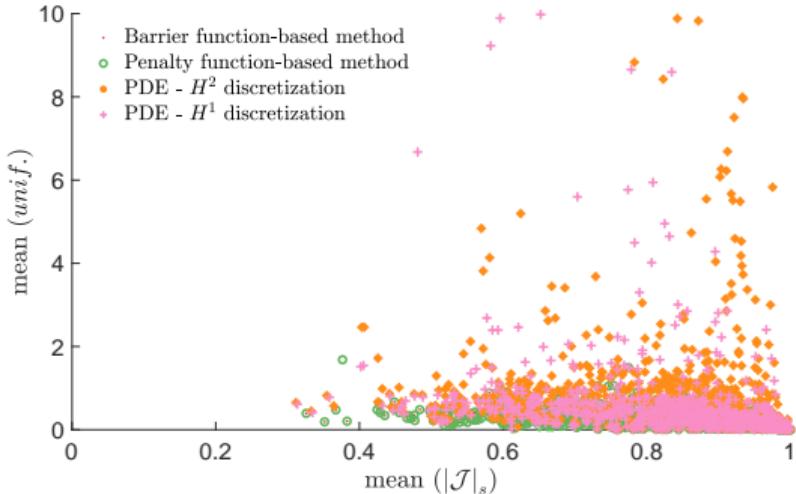
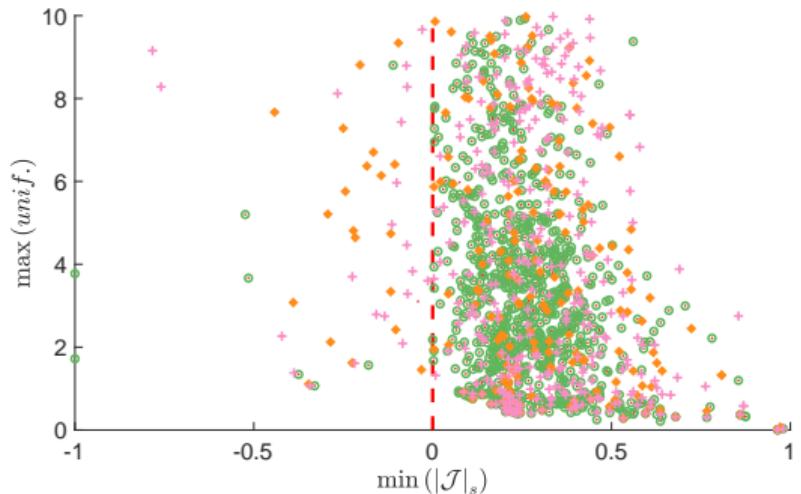
- PDE-based method: Excels in domains with extreme aspect ratios.
- **Real-world application:** Used in a rotary twin-screw compressor simulation.



Planar Parameterization Test Dataset (977 models)



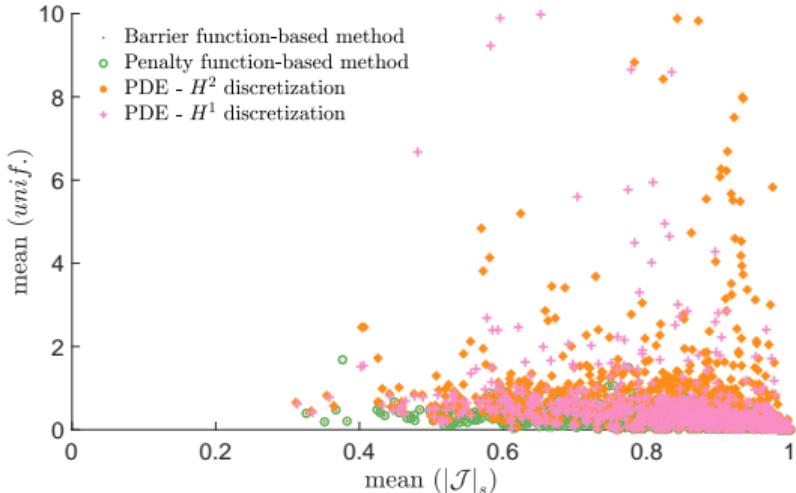
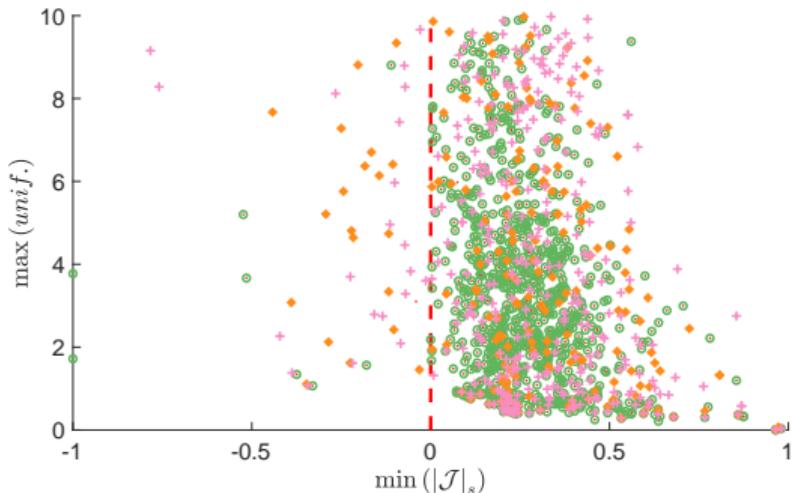
Effectiveness and Quality Assessment



Success rates:

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

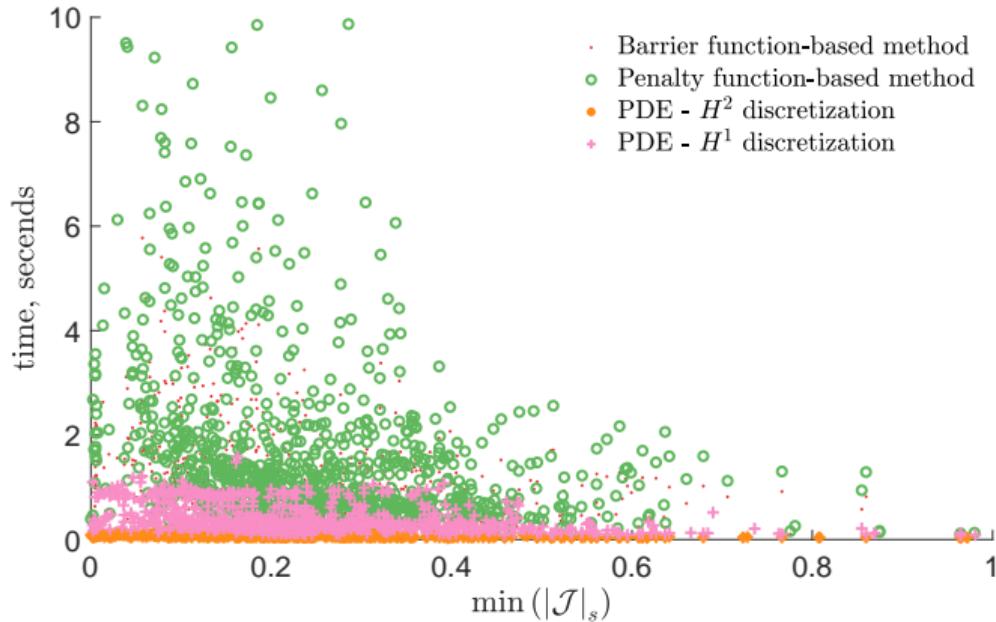
Effectiveness and Quality Assessment



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]: $961/977 \simeq \mathbf{98.36\%}$;
- 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

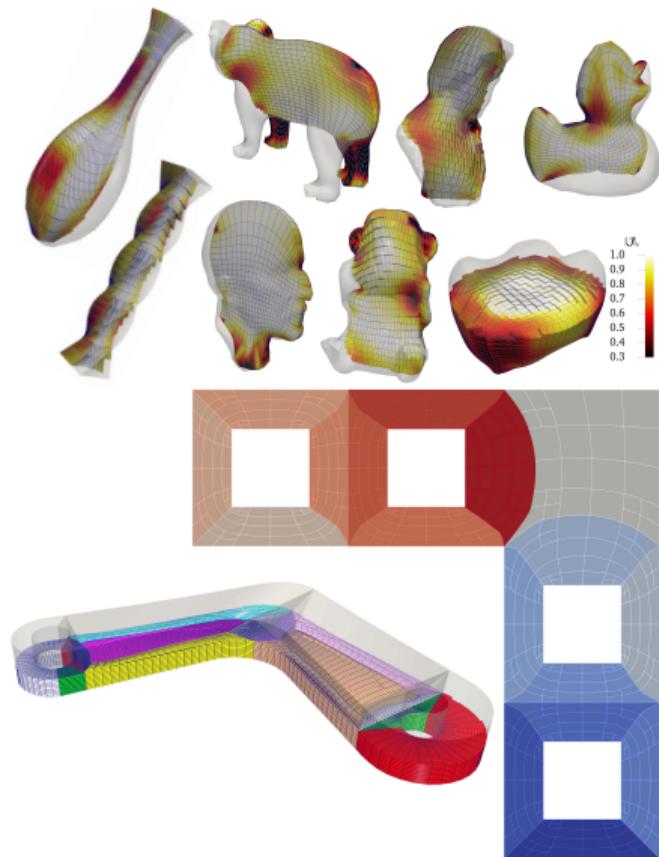
- ① Research Background and Motivation
- ② Parameterization Quality Improvement: Scaled H^1 Discretization
- ③ Enhancing Computational Efficiency: Preconditioned Anderson Acceleration
- ④ Results and Applications
- ⑤ Conclusions and Outlook

Conclusions and Outlook

- Adopted Sobolev space H^1 **discretization** to achieve uniform element distribution.
- Proposed a novel **preconditioned Anderson acceleration** framework.
- Provided an **open-source implementation** within the G+Smo platform.

Future Work:

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



Poster: Multi-patch parameterization method for IGA ...



Multi-patch parameterization method for IGA using singular structure of cross-field

Ye Ji, Yi Zhang, Chun-Gang Zhu

School of Mathematical Sciences, Dalian University of Technology

Problem Definition & Contribution

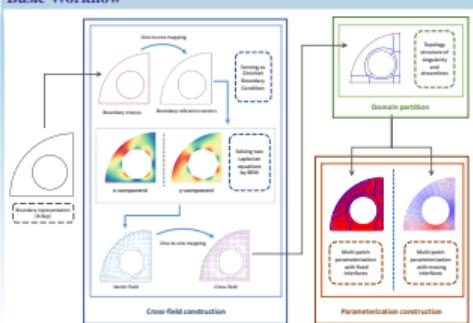
Goal: To construct an **internal spline-based parameterization** \mathbf{x} from the boundary representation (B-Rep), such that \mathbf{x} **ensures injectivity** and exhibits **minimal angle and area distortion**.



Key Contributions: A three-step strategy for constructing multi-patch parameterizations:

- **Cross-Field construction:** Utilize the one-to-one mapping between cross-field and vector-field;
- **Domain partition:** Analyze cross-field singular structure and propose a streamline propagation;
- **Multi-patch parameterization:** Develop two optimization-based methods.

Basic Workflow



Formulation

Cross-field construction:

- One-to-one mapping between a cross $v_k(p)$ and its reference vector $u(p)$ can be established as:

$$\begin{cases} \|u(p)\| = \|v_k(p)\|, \\ \theta_i(p) = 4 - \min\{\theta_k(p) : k = 1, 2, 3, 4\}. \end{cases}$$

- Governing equation propagates the reference vectors $u(p)$ from the boundaries to the interior:

$$\begin{cases} \nabla^2 u_i(p) = 0, & p \in \Omega, \\ u_i(p) = u_0(p), & p \in \partial\Omega, \end{cases} \quad (i = x, y).$$

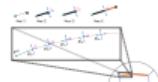
- **Boundary element method (BEM)** is employed since only the B-Rep is known.

Domain partition:

- Streamline propagation are calculated by:

$$p_{str}^k = p_{str}^{k-1} + \rho d\mathbf{r} \mathbf{V}^{k-1},$$

where $d\mathbf{r} \mathbf{V}^k = \frac{w_{ik}(P_{str}^k)}{\|w_{ik}(P_{str}^k)\|}, i_0 = \arg \min_{k \in \{1, 2, 3, 4\}} |\theta_k|$.



Multi-patch parameterization:

- Step 1 - Initialization: Begin with a Discrete Coons patch.
- Step 2 - Foldover elimination: Objective function

$$\delta^H = \int_{\Omega} \max \{0, \delta - |\mathcal{J}| \} \, d\Omega,$$

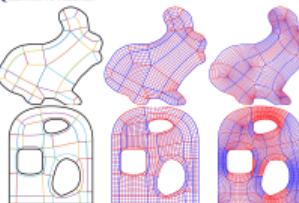
- Step 3 - Quality improvement: Objective function

$$\delta := \begin{cases} \int_{\Omega} \text{tr}(\mathcal{J}^T \mathcal{J}) / |\mathcal{J}| + \lambda |\mathcal{J}|^2 d\Omega, & \min |\mathcal{J}| > 0, \\ +\infty, & \min |\mathcal{J}| \leq 0. \end{cases}$$



Experiments & Results

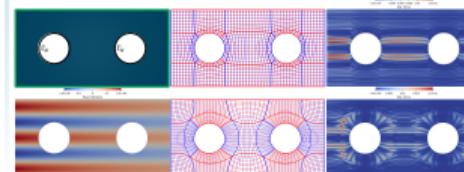
Qualitative results:



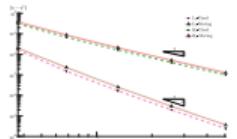
Quantitative results:

| Model | #Patch | Method | $\ \mathcal{J}\ _a$ | | | uni.f. | time (sec.) |
|---------|--------|-------------------|---------------------|--------|---------------|---------------|-----------------------------|
| | | | min | avg. | min | | |
| rabbit | 33 | fixed-interface | 0.2204 | 0.9584 | 0.6103 | 0.9544 | 0.9982 1.6169 |
| | | moving-interface | 0.02918 | 0.9283 | 0.0000 | 0.9550 | 1.0000 23.7138 |
| 2 holes | 21 | smoothness energy | -1 | 0.8353 | 0.6981 | 0.9082 | 0.10378 - |
| | | fixed-interface | 0.8466 | 0.9892 | 0.7354 | 0.9082 | 0.9875 0.2020 |
| 3 holes | 46 | moving-interface | 0.4782 | 0.9728 | 0.7096 | 0.9073 | 0.9674 0.5404 |
| | | discrete Coons | 0.5492 | 0.9710 | 0.8008 | 0.9573 | 1.0953 - |
| | | fixed-interface | 0.1545 | 0.9716 | 0.8007 | 0.9573 | 0.9978 0.4965 |
| | | moving-interface | 0.1461 | 0.9361 | 0.6791 | 0.9571 | 0.9968 5.8342 |

Solving Poisson equation using IGA:

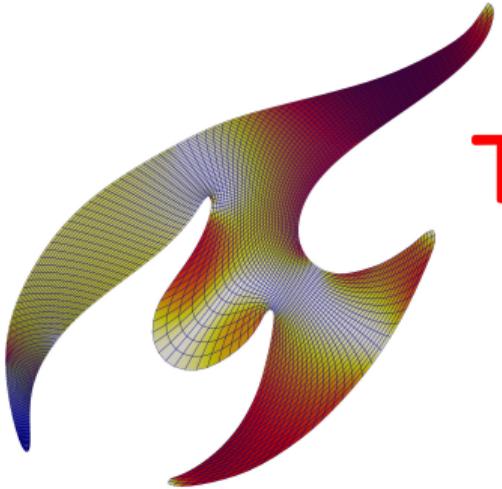


Convergence curves vs. DOF^{1/2}:



Please feel free to contact me! :-)

jyjess@outlook.com & <https://jyjess.github.io>



Thanks for Your Attention!

Q&A.

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

- Email: jiyess@outlook.com
- GitHub: [jiyess](https://github.com/jiyess)
- Homepage: <https://jiyess.github.io>

I am looking for a postdoctoral position!

Equivalence Problem: Unconstrained Optimization

- Recall the planar MIPS energy,

$$\begin{aligned}\mathcal{E}_{2D}^{\text{angle}}(\mathbf{x}) &= \frac{\sigma_1^2 + \sigma_2^2}{\sigma_1 \sigma_2} \\ &= \frac{\text{trace}(\mathcal{J}^T \mathcal{J})}{|\mathcal{J}|}.\end{aligned}$$

Since the Jacobian determinant appears in its denominator, it proceeds to infinity if the Jacobian determinant $|\mathcal{J}|$ approaches zero.

- Solve the following **unconstrained optimization problem**:

$$\arg \min_{\mathbf{P}_i, i \in \mathcal{I}_I} \mathcal{E}(\mathbf{x}) = \int_{\hat{\Omega}} \left(\lambda_{\text{angle}} \mathcal{E}^{\text{angle}}(\mathbf{x}) + \lambda_{\text{vol}} \mathcal{E}^{\text{vol}}(\mathbf{x}) \right) d\hat{\Omega}. \quad (1)$$

Local/Global Parameterization

