Computer Aided Geometric Design 102 (2023) 102191

Contents lists available at ScienceDirect

Computer Aided Geometric Design

journal homepage: www.elsevier.com/locate/cagd

On an improved PDE-based elliptic parameterization method for isogeometric analysis using preconditioned Anderson acceleration

Ye Ji [b,c,f], Kewang Chen [a,c,d,e,*], Matthias Möller [c], Cornelis Vuik [c]

[a] *College of Mathematics and Statistics, Nanjing University of Information Science and Technology, Nanjing, 210044, China*
[b] *School of Mathematical Sciences, Dalian University of Technology, Dalian, 116024, China*
[c] *Delft Institute of Applied Mathematics, Delft University of Technology, Delft, 2628CD, the Netherlands*
[d] *Jiangsu International Joint Laboratory on System Modeling and Data Analysis, Nanjing University of Information Science and Technology, Nanjing, 210044, China*
[e] *Center for Applied Mathematics of Jiangsu Province, Nanjing University of Information Science and Technology, Nanjing, 210044, China*
[f] *Key Laboratory for Computational Mathematics and Data Intelligence of Liaoning Province, Dalian, 116024, China*

ARTICLE INFO

Article history:
Available online 18 April 2023

Keywords:
Isogeometric analysis
Analysis-suitable parameterization
Anderson acceleration
Nonlinear preconditioning

ABSTRACT

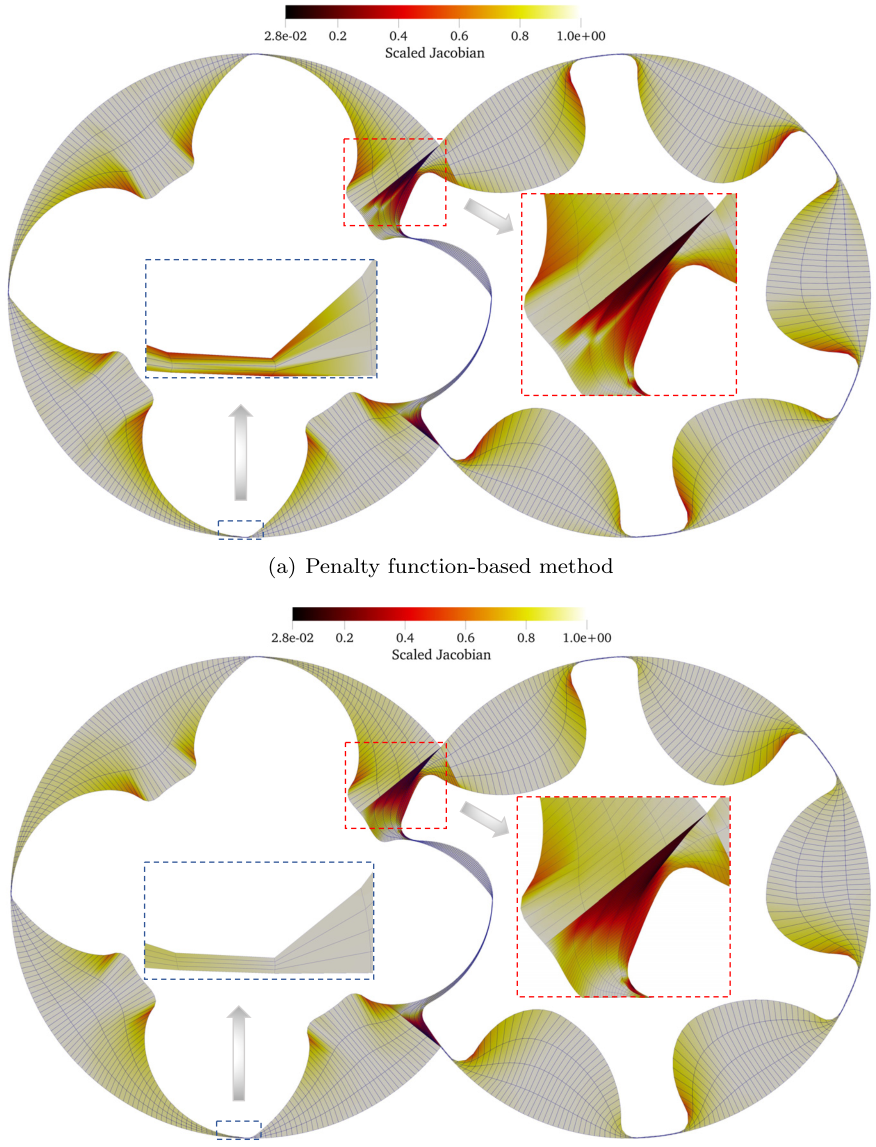
Constructing an analysis-suitable parameterization for the computational domain from its boundary representation plays a crucial role in the isogeometric design-through-analysis pipeline. PDE-based elliptic grid generation is an effective method for generating high-quality parameterizations with rapid convergence properties for the planar case. However, it may generate non-uniform grid lines, especially near the concave/convex parts of the boundary. In the present work, we introduce a novel scaled discretization of harmonic mappings in the Sobolev space H^1 to remit it. Analytical Jacobian matrices for the involved nonlinear equations are derived to accelerate the computation. To enhance the numerical stability and the speed of convergence, we propose a simple and yet effective preconditioned Anderson acceleration framework instead of using computationally expensive Newton-type iteration. Three preconditioning strategies are suggested, namely diagonal Jacobian, block-diagonal Jacobian, and full Jacobian. Furthermore, we discuss a delayed update strategy of the preconditioner, i.e., the preconditioner is updated every few steps to reduce the computational cost per iteration. Numerical experiments demonstrate the effectiveness and efficiency of our improved parameterization approach and the computational efficiency of our preconditioned Anderson acceleration scheme.

1. Introduction

The advent of Isogeometric Analysis (IGA) Hughes et al. (2005) sheds light on bridging the gap between Computer-Aided Design (CAD) and downstream Computer-Aided Engineering (CAE). Compared with the conventional finite element method (FEM), IGA employs the same spline-based basis functions in geometry modeling and numerical simulation. It avoids redundant data type conversions from spline-based CAD models to linear mesh models and thus eliminates geometry errors during the subsequent analysis stage.

[*] Corresponding author at: College of Mathematics and Statistics, Nanjing University of Information Science and Technology, Nanjing, 210044, China.
E-mail addresses: jiye@mail.dlut.edu.cn (Y. Ji), kwchen@nuist.edu.cn (K. Chen), m.moller@tudelft.nl (M. Möller), cvuik@tudelft.nl (C. Vuik).

https://doi.org/10.1016/j.cagd.2023.102191
0167-8396/© 2023 Elsevier B.V. All rights reserved.



(a) Penalty function-based method

(b) Elliptic Grid Generation (EGG) method

Fig. 1. Parameterizations generated by penalty function based method Ji et al. (2022b) and EGG method Hinz (2020). (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

However, most modern CAD systems describe the geometries of products using the so-called boundary representation (B-Rep) Cohen et al. (2010). Constructing an analysis-suitable spline-based parameterization for the computational domain from its B-Rep constitutes the first and essential step in the isogeometric pipeline. The task of generating an analysis-suitable parameterization consists in constructing a spline-based mapping $\mathbf{x}(\boldsymbol{\xi})$ from the parametric domain $\hat{\Omega}$ to the computational domain Ω. For an analysis-suitable parameterization, its bijectivity always comes first, then it should minimize angle and area distortion. Xu et al. (2013c); Pilgerstorfer and Jüttler (2014) reveal that parameterization quality significantly affects the accuracy and efficiency of the subsequent analysis.

Harmonic mapping theory gains much attention for planar parameterizations owing to its good mathematical properties and solid theoretical foundations. Various computational techniques for approximating the inverse of a harmonic mapping have been developed, such as the variational harmonic method Xu et al. (2013b), least-squares fitting Nguyen and Jüttler (2010); Falini et al. (2015), minimizing (modified) Winslow's function Gravesen et al. (2012); Ji et al. (2021), and the PDE-based elliptic grid generation (EGG) approach Hinz et al. (2018a); Hinz (2020).

Although all the above methods attempt to approximate the inverse harmonic mapping in finite-dimensional spline spaces, their numerical behavior can be quite different. Fig. 1 shows a slice of a rotary twin-screw compressor with a challenging geometry, particularly the extreme aspect ratios between the clearance of both rotors. This geometry is from a real-world use case. A broad range of parameterization techniques shows good performance on many established benchmark geometries, while then failing when utilized in real-world applications, unfortunately. The parameterization produced by the recently developed penalty-function based method Ji et al. (2022b) is shown in Fig. 1(a). For this challenging geometry, an appropriate selection of the relevant setting parameters becomes essential. In addition, the size of the parametric domain significantly affects the parameterization result, which is undesirable as this is an artificial parameter that needs to be set by the user. By contrast, the PDE-based EGG method Hinz et al. (2018a) produces a well-pleasing parameterization, as shown in Fig. 1(b).

In the PDE-based method, one needs to solve a coupled system of nonlinear equations, and thus the Newton-type solvers are suggested by Hinz et al. (2018a). However, the Jacobian matrix or a function that implements its application to a vector (so-called matrix-free Newton-Krylov methods) is needed during nonlinear iterations, which may be quite expensive for large-scale problems.

In 1962, Anderson (1965, 2019) developed a technique called extrapolation algorithm for accelerating the convergence of nonlinear fixed-point iterations. This technique is now called Anderson acceleration (AA) in the applied mathematics community. Unlike Newton-type methods, one advantage of AA is that it does not require the expensive computation or approximation of the Jacobian matrices.

In this paper, we propose several efficient solving strategies for the PDE-based EGG parameterization method. The basic idea is to convert the nonlinear problem into a fixed-point iteration problem and then apply (preconditioned) AA to speed up the convergence. Our main contributions are as follows:

- One disadvantage of EGG is that some nonuniform grid elements may occur, in particular, near the convex part and the concave part of the boundary. To weaken its impact and improve the quality of the parameterization, we develop a novel scaled version of harmonic maps in the Sobolev space H^1.
- Jacobian matrices are of vital importance during nonlinear iterations. To enhance the computational efficiency and the numerical stability, we furthermore derive analytical Jacobian matrices for both discretizations in the Sobolev space H^2 and H^1.
- AA algorithm is adopted to avoid the calculation of Jacobian matrix and reduce computational overhead. To improve its robustness and accelerate convergence, we propose a framework of preconditioned AA schemes. Three preconditioning strategies are proposed, namely diagonal Jacobian preconditioner, block-diagonal Jacobian preconditioner, and full Jacobian preconditioner. To reduce the computational cost per iteration, we update the preconditioner for AA every few steps. Numerical experiments show the effectiveness of the proposed update strategy.

2. Related work

This section reviews the related works on parameterizations for IGA and Anderson acceleration.

2.1. Analysis-suitable parameterization construction for IGA

Before leading into isogeometric simulation, the foremost step is to construct an analysis-suitable parameterization. Cohen et al. (2010); Xu et al. (2013c); Pilgerstorfer and Jüttler (2014) point out the profound role of the parameterization quality in the downstream analysis. The first criterion of high-quality parameterizations is bijectivity. In addition to bijectivity, low angle and area distortion is preferred. In this section, we classify the existing approaches by the way to treat the bijectivity constraints.

1) *Algebraic methods* are those that solve one linear system at most. A typical representative is the discrete Coons method Farin and Hansford (1999), a particular transfinite interpolation (TFI). Gravesen et al. (2012) introduce several linear parameterization methods, such as the spring patch and the mean value coordinates, where one linear system is solved. These methods win in efficiency but lose in robustness. In particular, they tend to fold for complex domains. Thus they are typically employed for constructing an initial guess only in the parameterization pipeline.
2) *Nonlinear constrained optimization methods* are based on the observation that the Jacobian determinant of the parameterization can be expressed by a higher-order NURBS function. Xu et al. (2011) set the nonlinear coefficients as inequality constraints to guarantee the bijectivity. Subsequently, Xu et al. (2013c) generalize this method to the volumetric case. By using a divide-and-conquer strategy, Wang and Qian (2014) present an accelerated optimization framework. To topology-consistent domains, Xu et al. (2017) introduce a computation reuse method. Ugalde et al. (2018) propose some sufficient conditions and a necessary condition for the bijectivity of biquadratic B-splines. Pan and Chen (2019) develop a low-rank method for volumetric parameterization problems. However, the number of inequality constraints is usually quite large, which is impractical for large-scale problems. To this end, Pan et al. (2020) propose a collocation strategy to reduce the computational burden. Xu et al. (2019); Ji et al. (2022c) enhance the numerical accuracy by minimizing the curvature metric of the IGA solution surface.
3) *Harmonic mapping-based methods* become popular recently as they may avoid the arduous calculation burden of nonlinear constraints. All of these types of methods attempt to approximate the stationary point of the well-known Dirichlet

energy satisfying given boundary conditions or its corresponding Euler-Lagrange equation. In this sense, the existing methods can be roughly classified into two categories.

The first type of methods minimizes Winslow's function, also known as Most-Isometric ParameterizationS (MIPS) energy in the computer graphics community Hormann and Greiner (2000). However, these methods need an already bijective initial guess since the Jacobian determinant appears in the denominator position of the Winslow's function. To this end, different foldover elimination manners are proposed. Su et al. (2019); Liu et al. (2020) project the Jacobian matrix to the K-bounded distortion space. Recently, Zheng and Chen (2022) apply a similar idea to THB-spline parameterizations. Ji et al. (2021) eliminate foldovers efficiently by solving a simple unconstrained optimization problem. Another idea is to use a penalty function and Jacobian regularization technique, which dates back to Garanzha and Kaporin (1999); Garanzha et al. (2021) in the mesh untangling problem. Wang and Ma (2021) adopt this idea to avoid extra foldover elimination steps. Ji et al. (2022b) propose a new penalty term to remove the numerical error of the previous penalty term. Apart from Winslow's functional, some works follow the quasi-conformal theory, such as the Teichmüller mapping Nian and Chen (2016) and the low-rank quasi-conformal method Pan et al. (2018).

The second type of methods approximates the Laplace equation with Dirichlet boundary conditions. Martin et al. (2009) adopt discrete harmonic functions for trivariate B-spline solids. Nguyen and Jüttler (2010) construct a parameterization using a sequence of harmonic maps. Xu et al. (2013b) develop a variational harmonic mapping method. Falini et al. (2015) compute a harmonic map from a physical domain to a parametric domain by boundary element method and then approximate its inverse map by least-squares fitting. Based on the principle of Elliptic Grid Generation (EGG), Hinz et al. (2018a,b); Hinz (2020) propose a PDE-based method for IGA-suitable parameterizations.

What is noteworthy is that the PDE-based EGG method has a nice convergence property and performs particularly well in computational domains with extreme aspect ratios, see Fig. 1(b). However, there are two main issues with it. First, it usually produces nonuniform elements near the concave/convex parts of the boundary for general domains. Second, the standard Newton method as adopted in Hinz (2020) is inefficient for large-scale problems. This paper devotes attention to an efficient solver named preconditioned Anderson acceleration for the PDE-based EGG method and to an improvement of the parameterization quality for general computational domains by introducing a scaled version in H^1 space.

To complex computational domains, particularly high-genus domains, single patch parameterization may be insufficient. Multi-patch and multi-block techniques are widely adopted to decompose a given domain into several simple regions. Interested readers are referred to Xu et al. (2013a, 2015, 2018); Buchegger and Jüttler (2017); Xiao et al. (2018); Falini and Jüttler (2019); Haberleitner et al. (2019); Chen et al. (2019, 2022); Zhang et al. (2021); Bastl and Slabá (2021); Wang et al. (2022a); Shepherd et al. (2022a,b) and the references therein.

In addition, some works consider parameterization techniques using non-standard B-splines, such as T-splines Zhang et al. (2012, 2013), (truncated) HB-splines Falini et al. (2015); Pan and Chen (2022); Zheng and Chen (2022), toric patches Ji et al. (2022a), PHT-splines Chan et al. (2017), and subdivision methods Pan et al. (2021); Xie et al. (2020), and TCB-splines Wang et al. (2022b).

2.2. *Anderson acceleration (AA)*

Anderson acceleration methods are considered "essentially equivalent" to the nonlinear GMRES methods Carlson and Miller (1998); Oosterlee and Washio (2000); Walker and Ni (2011) and the direct inversion on the iterative subspace method Pulay (1980); Lin and Yang (2013). They are also in a broad category with methods based on quasi-Newton updating Fang and Saad (2009); Haelterman et al. (2010). On linear problems, for instance, Walker and Ni (2011) prove that AA without restarting ($m = \infty$) is equivalent in a certain sense to the GMRES method. Fang and Saad (2009) have shown a remarkable relation between AA and quasi-Newton methods for solving nonlinear equations, which utilizes the previous iterates to approximate the inverse Jacobian. The behavior and the potential of the non-stationary AA with dynamic window sizes have not been studied intensively. Here, the window sizes mean how many previous iterates are used in AA. For now, only a few results are available. Evans et al. (2020) propose a heuristic strategy to choose the damping factors based on the gain at each iteration. Pollock and Rebholz (2021) develop a strategy to alternate the window sizes dynamically. More recently, motivated by the hybrid linear solver GMRESR (GMRES Recursive) Vuik (1993); Van der Vorst and Vuik (1994), Chen and Vuik (2022a) propose a composite Anderson acceleration method with two window sizes.

Convergence acceleration by AA has been widely observed, the convergence analysis, however, has been reported only recently. In 2015, Toth and Kelley (2015) first prove that the stationary version of AA (sAA) without damping is locally r-linearly convergent if the fixed point map is a contraction. Later, Evans et al. (2020) extend the result to AA with damping factors. Recently, Pollock et al. (2019) apply sAA to the Picard iteration for solving the steady incompressible Navier–Stokes equations and prove that the acceleration improves the convergence rate of the Picard iteration. Sterck and He (2021) extend the result to a more general fixed-point iteration of the form $\mathbf{x} = \mathbf{Q}(\mathbf{x})$, given knowledge of the spectrum of $\mathbf{Q}'(\mathbf{x})$ at fixed-point $\mathbf{x}^*$. Besides, Wang et al. (2021) study the asymptotic linear convergence speed of sAA applied to the Alternating Direction Method of Multipliers (ADMM) method. Global convergence and sharper local convergence results of AA remain active research topics. For more related results about AA and its applications, we refer interested readers to the papers by Sterck (2012); Brune et al. (2015); Toth et al. (2017); Peng et al. (2018); Zhang et al. (2019, 2020); Bian et al. (2021); Chen and Vuik (2022b) and the references therein.

3. Problem statement and notations

This section is devoted to introduce our problem statement and describe the notations.

3.1. Problem statement

Mathematically, a planar parameterization is a spline-based mapping $\boldsymbol{x}$ from the parametric domain $\hat{\Omega}$ to the computational domain Ω, i.e.,

$$\boldsymbol{x}(\boldsymbol{\xi}) = \begin{pmatrix} x(\boldsymbol{\xi}) \\ y(\boldsymbol{\xi}) \end{pmatrix} = \sum_{i_1=0}^{n_1} \sum_{i_2=0}^{n_2} \mathbf{P}_{i_1,i_2} R_{i_1,i_2}(\boldsymbol{\xi}), \quad \boldsymbol{\xi} = (\xi, \eta) \in \hat{\Omega}, \tag{1}$$

where

$$R_{i_1,i_2}(\xi,\eta) = \frac{\omega_{i_1,i_2} N_{i_1}^{p_1}(\xi) N_{i_2}^{p_2}(\eta)}{\sum_{i_1=0}^{n_1} \sum_{i_2=0}^{n_2} \omega_{i_1,i_2} N_{i_1}^{p_1}(\xi) N_{i_2}^{p_2}(\eta)} \tag{2}$$

are bivariate tensor-product NURBS basis functions of bi-degree (p_1, p_2), ω_{i_1,i_2} are weights, and $\mathbf{P}_{i_1,i_2} \in \mathbb{R}^2$ are the corresponding control points.

Denote by $\mathcal{I}_I$ and $\mathcal{I}_B$ the index set of the unknown inner control points and the known boundary control points, respectively. Then the parameterization $\boldsymbol{x}$ in (1) can be split into two partial sums

$$\boldsymbol{x}(\boldsymbol{\xi}) = \mathbf{R}^{\mathrm{T}}\mathbf{P} = \underbrace{\sum_{i \in \mathcal{I}_I} \mathbf{P}_i R_i(\boldsymbol{\xi})}_{\text{unknown}} + \underbrace{\sum_{j \in \mathcal{I}_B} \mathbf{P}_j R_j(\boldsymbol{\xi})}_{\text{known}}, \tag{3}$$

where $\mathbf{R}$ and $\mathbf{P}$ represent the column collections of the NURBS basis functions and the control points respectively, $\mathbf{P}_i$ are unknown inner control points, and $\mathbf{P}_j$ are the given boundary control points.

Our parameterization problem can be stated as follows: construct the unknown inner control points $\mathbf{P}_i$, $i \in \mathcal{I}_I$ such that the parameterization $\boldsymbol{x}$ is bijective and has low distortions.

3.2. Notations

First, denote by

$$\nabla_{\boldsymbol{\xi}}\mathbf{R} = \begin{bmatrix} R_{0,\xi}, & R_{1,\xi}, & \ldots, & R_{n,\xi} \\ R_{0,\eta}, & R_{1,\eta}, & \ldots, & R_{n,\eta} \end{bmatrix} \tag{4}$$

and

$$\nabla_{\boldsymbol{x}}\mathbf{R} = \begin{bmatrix} R_{0,x}, & R_{1,x}, & \cdots, & R_{n,x} \\ R_{0,y}, & R_{1,y}, & \cdots, & R_{n,y} \end{bmatrix} \tag{5}$$

the partial derivatives of the basis functions w.r.t. the parameter coordinates $\boldsymbol{\xi}$ and the physical coordinates $\boldsymbol{x}$, respectively. Then the Jacobian matrix of the parameterization $\boldsymbol{x}$ can be represented as

$$\mathcal{J} = \begin{bmatrix} x_\xi & y_\xi \\ x_\eta & y_\eta \end{bmatrix} = \nabla_{\boldsymbol{\xi}}\mathbf{R}\,\mathbf{P}. \tag{6}$$

We denote by $|\mathcal{J}|$ and $\mathcal{J}^{-1}$ its determinant and the inverse of the Jacobian matrix, respectively.

According to the transformation relation between two coordinates systems, we have that

$$\nabla_{\boldsymbol{x}}\mathbf{R} = \mathcal{J}^{-1}\nabla_{\boldsymbol{\xi}}\mathbf{R}. \tag{7}$$

In addition, the first metric tensor

$$\boldsymbol{g} = \begin{bmatrix} g_{11} & g_{12} \\ g_{12} & g_{22} \end{bmatrix} = \begin{bmatrix} \boldsymbol{x}_\xi \cdot \boldsymbol{x}_\xi & \boldsymbol{x}_\xi \cdot \boldsymbol{x}_\eta \\ \boldsymbol{x}_\xi \cdot \boldsymbol{x}_\eta & \boldsymbol{x}_\eta \cdot \boldsymbol{x}_\eta \end{bmatrix} = \mathcal{J}\mathcal{J}^{\mathrm{T}} \tag{8}$$

is of vital importance in characterizing the geometric properties of $\boldsymbol{x}$.

4. Elliptic parameterization method

4.1. Basic principle of elliptic grid generation

Elliptic grid generation (EGG) is a commonly used technique for generating structured grids from a given boundary description of the computational domain. Hinz et al. (2018a) introduce this method into computing IGA-suitable parameterization.

The basic idea of EGG is to compute a harmonic mapping $\boldsymbol{x}$ between the parametric domain $\hat{\Omega}$ and the computational domain Ω by solving the following Laplace equations:

$$\begin{cases} \Delta\xi(x,y)=0 \\ \Delta\eta(x,y)=0 \end{cases} \quad \text{s.t. } \boldsymbol{x}^{-1}|_{\partial\Omega}=\partial\hat{\Omega}. \tag{9}$$

The problem (9) is a particular class of Dirichlet problems whose solution exists if $\partial\Omega$ satisfies $C^{1,\alpha}$ Hölder continuity condition for some $\alpha\in(0,1)$, and the uniqueness of the solution is guaranteed by the maximum principle. Since the parametric domain $\hat{\Omega}$ is assumed to be convex (usually a unit square), the unique solution $\boldsymbol{x}^{-1}$ offers a one-to-one mapping with the Jacobian $\mathcal{J}$ not vanishing between the interior of the parametric domain $\hat{\Omega}$ and the interior of the computational domain Ω, which is ensured by the Radó–Kneser–Choquet theorem Duren and Hengartner (1997).

According to the first variation formula, solving the equations $\Delta\boldsymbol{\xi}=0$ with the specified boundary conditions in (9) is equivalent to finding a mapping $\boldsymbol{x}^{-1}$ that satisfies the same boundary conditions with minimal Dirichlet energy

$$E_D=\frac{1}{2}\int_{\Omega}\left(\left(\frac{\partial\xi}{\partial x}\right)^2+\left(\frac{\partial\xi}{\partial y}\right)^2+\left(\frac{\partial\eta}{\partial x}\right)^2+\left(\frac{\partial\eta}{\partial y}\right)^2\right)d\Omega. \tag{10}$$

Switching the integration in (10) from Ω to $\hat{\Omega}$ yields

$$E_D=\frac{1}{2}\int_{\hat{\Omega}}\frac{x_\xi^2+x_\eta^2+y_\xi^2+y_\eta^2}{|\mathcal{J}|}d\hat{\Omega}, \tag{11}$$

which is the well-known Winslow's function. Many parameterization methods are proposed by minimizing the above Winslow's function and its variants, e.g., Gravesen et al. (2012); Ji et al. (2021).

It means that the problem (9) and finding the minimum of Winslow's function (11) are essentially equivalent in Sobolev space H^1. What is particularly interesting is that, in a finite-dimensional spline space, the parameterizations produced by these two methods can be radically different (see again Fig. 1).

4.2. Discretization in Sobolev space H^2

In the context of generating parameterizations for IGA, one is more interested in the mapping $\boldsymbol{x}$ from $\hat{\Omega}$ to Ω, which is the inverse of harmonic mapping $\boldsymbol{x}^{-1}$. To this end, the set of Laplace equations (9) is converted to its equivalent problem Xu et al. (2013b). The resulting nonlinear vector-valued second-order PDE problem reads:

$$\begin{cases} \mathcal{L}x=0 \\ \mathcal{L}y=0 \end{cases} \quad \text{s.t. } \boldsymbol{x}|_{\partial\hat{\Omega}}=\partial\Omega, \tag{12}$$

where

$$\mathcal{L}=g_{22}\frac{\partial^2}{\partial\xi^2}-2g_{12}\frac{\partial^2}{\partial\xi\partial\eta}+g_{11}\frac{\partial^2}{\partial\eta^2} \tag{13}$$

is a differential operator and g_{ij} denotes the entries of the metric tensor $\mathbf{g}$ in (8).

To have a better convergence, Hinz et al. (2018b) replace the differential operator $\mathcal{L}$ in (13) by

$$\widetilde{\mathcal{L}}=\frac{\mathcal{L}}{g_{11}+g_{22}}. \tag{14}$$

The above operator (14) has a more consistent convergence criterion to various length-scaled geometries and a better convergence property in numerical experiments. Therefore, we follow this scheme in this paper.

Denote by Σ the spline space spanned by the NURBS basis functions in (3). Let $\Sigma_0=\{R_i\in\Sigma : R_i|_{\partial\hat{\Omega}}=0\}$ be the collection of $R_i\in\Sigma$ that vanish on $\partial\hat{\Omega}$. Following the IGA setting, we have the following variational counterpart of (7)

$$\forall R_i\in\Sigma_0 : \begin{cases} \mathbf{F}^x=\mathbf{0}, \\ \mathbf{F}^y=\mathbf{0}, \end{cases} \quad \text{s.t. } \boldsymbol{x}|_{\partial\hat{\Omega}}=\partial\Omega, \tag{15}$$

where

$$\mathbf{F}^x = \int\limits_{\hat{\Omega}} \mathbf{R}\, \widetilde{\mathcal{L}}x \, d\hat{\Omega}, \tag{16}$$

$$\mathbf{F}^y = \int\limits_{\hat{\Omega}} \mathbf{R}\, \widetilde{\mathcal{L}}y \, d\hat{\Omega}, \tag{17}$$

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

Then the unknown inner control points can be obtained by solving the above nonlinear system where the known boundary control points serve as Dirichlet boundary conditions.

4.3. Analytical Jacobian computation for the discretization in H^2 space

The analytical Jacobian has a crucial influence not only on the overall computational efficiency but also on the numerical stability. In this section, we introduce the calculation of the analytical Jacobian matrix for the nonlinear system (15).

For the clarity of notations, we denote by $\partial_{\mathbf{P}^x} = \partial/\partial\mathbf{P}^x$ and $\partial_{\mathbf{P}^y} = \partial/\partial\mathbf{P}^y$ the partial derivatives w.r.t. all the x-coordinate components and all the y-coordinate components of the control points $\mathbf{P}$. In addition, let us denote by $()_{,\alpha}$ and $()_{,\alpha\beta}$ ($\alpha = \xi, \eta$ and $\beta = \xi, \eta$) the first and the second derivatives over the parameters.

First, by taking the derivatives of the entries g_{ij} of the metric tensor, we have

$$\begin{aligned}
&\partial_{\mathbf{P}^x} g_{11} = 2x_\xi \mathbf{R}_{,\xi}, && \partial_{\mathbf{P}^y} g_{11} = 2y_\xi \mathbf{R}_{,\xi},\\
&\partial_{\mathbf{P}^x} g_{22} = 2x_\eta \mathbf{R}_{,\eta}, && \partial_{\mathbf{P}^y} g_{22} = 2y_\eta \mathbf{R}_{,\eta},\\
&\partial_{\mathbf{P}^x} g_{12} = x_\eta \mathbf{R}_{,\xi} + x_\xi \mathbf{R}_{,\eta}, && \partial_{\mathbf{P}^y} g_{12} = y_\eta \mathbf{R}_{,\xi} + y_\xi \mathbf{R}_{,\eta}.
\end{aligned} \tag{18}$$

Through the product rules for derivatives and inserting the above formulas, we obtain

$$\begin{aligned}
\partial_{\mathbf{P}^x} \mathcal{L}x &= x_{,\xi\xi} \partial_{\mathbf{P}^x} g_{22} - 2x_{,\xi\eta} \partial_{\mathbf{P}^x} g_{12} + x_{,\eta\eta} \partial_{\mathbf{P}^x} g_{11} + g_{22}\mathbf{R}_{,\xi\xi} - 2g_{12}\mathbf{R}_{,\xi\eta} + g_{11}\mathbf{R}_{,\eta\eta},\\
\partial_{\mathbf{P}^y} \mathcal{L}y &= y_{,\xi\xi} \partial_{\mathbf{P}^y} g_{22} - 2y_{,\xi\eta} \partial_{\mathbf{P}^y} g_{12} + y_{,\eta\eta} \partial_{\mathbf{P}^y} g_{11} + g_{22}\mathbf{R}_{,\xi\xi} - 2g_{12}\mathbf{R}_{,\xi\eta} + g_{11}\mathbf{R}_{,\eta\eta},\\
\partial_{\mathbf{P}^y} \mathcal{L}x &= x_{,\xi\xi} \partial_{\mathbf{P}^y} g_{22} - 2x_{,\xi\eta} \partial_{\mathbf{P}^y} g_{12} + x_{,\eta\eta} \partial_{\mathbf{P}^y} g_{11},\\
\partial_{\mathbf{P}^x} \mathcal{L}y &= y_{,\xi\xi} \partial_{\mathbf{P}^x} g_{22} - 2y_{,\xi\eta} \partial_{\mathbf{P}^x} g_{12} + y_{,\eta\eta} \partial_{\mathbf{P}^x} g_{11}.
\end{aligned} \tag{19}$$

Next, the partial derivatives of the scaled differential operator (14) can be computed as follows

$$\begin{aligned}
\partial_{\mathbf{P}^x} \widetilde{\mathcal{L}}x &= \frac{\partial_{\mathbf{P}^x} \mathcal{L}x - \widetilde{\mathcal{L}}x\,(\partial_{\mathbf{P}^x} g_{11} + \partial_{\mathbf{P}^x} g_{22})}{g_{11} + g_{22}}, & \partial_{\mathbf{P}^y} \widetilde{\mathcal{L}}x &= \frac{\partial_{\mathbf{P}^y} \mathcal{L}x - \widetilde{\mathcal{L}}x\,(\partial_{\mathbf{P}^y} g_{11} + \partial_{\mathbf{P}^y} g_{22})}{g_{11} + g_{22}},\\
\partial_{\mathbf{P}^x} \widetilde{\mathcal{L}}y &= \frac{\partial_{\mathbf{P}^x} \mathcal{L}y - \widetilde{\mathcal{L}}y\,(\partial_{\mathbf{P}^x} g_{11} + \partial_{\mathbf{P}^x} g_{22})}{g_{11} + g_{22}}, & \partial_{\mathbf{P}^y} \widetilde{\mathcal{L}}y &= \frac{\partial_{\mathbf{P}^y} \mathcal{L}y - \widetilde{\mathcal{L}}y\,(\partial_{\mathbf{P}^y} g_{11} + \partial_{\mathbf{P}^y} g_{22})}{g_{11} + g_{22}}.
\end{aligned} \tag{20}$$

Finally, the Jacobian matrix for the nonlinear system (15) is given by

$$jac(\mathbf{F}) = \begin{bmatrix} \partial_{\mathbf{P}^x}\mathbf{F}^x & \partial_{\mathbf{P}^x}\mathbf{F}^y \\ \partial_{\mathbf{P}^y}\mathbf{F}^x & \partial_{\mathbf{P}^y}\mathbf{F}^y \end{bmatrix}, \tag{21}$$

where

$$\begin{aligned}
\partial_{\mathbf{P}^x}\mathbf{F}^x &= \int\limits_{\hat{\Omega}} \mathbf{R}\partial_{\mathbf{P}^x} \widetilde{\mathcal{L}}x \, d\hat{\Omega}, & \partial_{\mathbf{P}^y}\mathbf{F}^x &= \int\limits_{\hat{\Omega}} \mathbf{R}\partial_{\mathbf{P}^y} \widetilde{\mathcal{L}}x \, d\hat{\Omega},\\
\partial_{\mathbf{P}^x}\mathbf{F}^y &= \int\limits_{\hat{\Omega}} \mathbf{R}\partial_{\mathbf{P}^x} \widetilde{\mathcal{L}}y \, d\hat{\Omega}, & \partial_{\mathbf{P}^y}\mathbf{F}^y &= \int\limits_{\hat{\Omega}} \mathbf{R}\partial_{\mathbf{P}^y} \widetilde{\mathcal{L}}y \, d\hat{\Omega}.
\end{aligned} \tag{22}$$

4.4. Discretization in Sobolev space H^1

The parameterization produced by the nonlinear system (15) is shown in the left of Fig. 2. One can see that some non-uniform elements appear near the duck's head and belly, highlighted by a red circle. As claimed in Hinz et al. (2018a), this issue can be partially remitted by performing refinements on the current geometry to better approximate harmonic mapping. However, these refinement operations introduce needless control points and make the CAD geometries heavy, which may lead to some troubles in the subsequent analysis and other downstream processes. This phenomenon is an

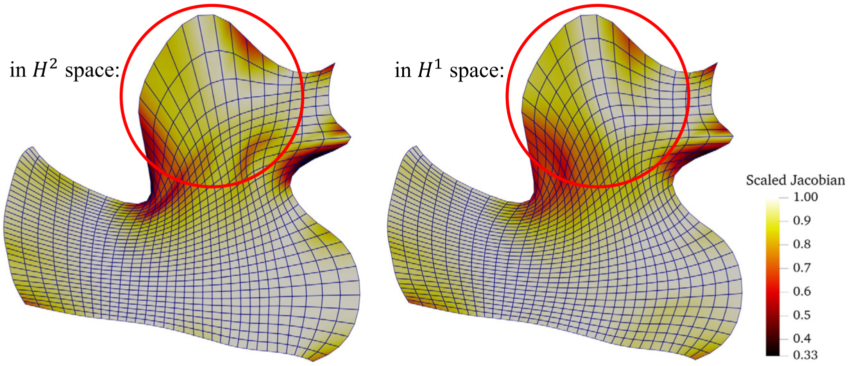


Fig. 2. Duck example: the left shows the parameterization produced by the discretization (15) in the H^2 space, where some non-uniform elements can be observed inside the red circle. The right shows the result produced by the discretization (30) in the H^1 space. The color encodes the scaled Jacobian, with white representing optimal orthogonality.

inherent property of EGG that has been widely observed Xu et al. (2013b). Now, we expect to improve the quality of the parameterization while leaving the cardinality of the control points fixed. In this section, we introduce a scale factor and present a novel discretization for (9) in the Sobolev space H^1 rather than H^2 to address this problem.

Consider the first component of the system of Laplace equations (9) and multiply it by the test function $R_i \in \Sigma_0$, then we have

$$R_i \Delta \xi = 0. \tag{23}$$

According to the product rule of divergence, the following equation holds

$$\nabla_{\boldsymbol{x}} \cdot (R_i \nabla_{\boldsymbol{x}} \xi) - \nabla_{\boldsymbol{x}} R_i \cdot \nabla_{\boldsymbol{x}} \xi = R_i \Delta \xi = 0. \tag{24}$$

For better scalability, here, we introduce a scaling by dividing the above by the Jacobian determinant

$$\frac{\nabla_{\boldsymbol{x}} \cdot (R_i \nabla_{\boldsymbol{x}} \xi) - \nabla_{\boldsymbol{x}} R_i \cdot \nabla_{\boldsymbol{x}} \xi}{|\mathcal{J}|} = 0. \tag{25}$$

Integrating the above equation (25) over the computational domain Ω and transforming it to the parametric domain $\hat{\Omega}$, we obtain

$$\begin{aligned} 0 &= \int\limits_{\Omega} \frac{\nabla_{\boldsymbol{x}} \cdot (R_i \nabla_{\boldsymbol{x}} \xi) - \nabla_{\boldsymbol{x}} R_i \cdot \nabla_{\boldsymbol{x}} \xi}{|\mathcal{J}|} \, d\Omega \\ &= \int\limits_{\hat{\Omega}} \nabla_{\boldsymbol{x}} \cdot (R_i \nabla_{\boldsymbol{x}} \xi) - \nabla_{\boldsymbol{x}} R_i \cdot \nabla_{\boldsymbol{x}} \xi \, d\hat{\Omega} \end{aligned} \tag{26}$$

According to Green's formula and exploiting the vanishing of $R_i \in \Sigma_0$ on $\partial\Omega$, we have

$$\int\limits_{\hat{\Omega}} \nabla_{\boldsymbol{x}} \cdot (R_i \nabla_{\boldsymbol{x}} \xi) d\hat{\Omega} = \oint\limits_{\hat{\Gamma}} R_i \nabla_{\boldsymbol{x}} \xi \cdot \mathbf{n} \, d\hat{\Gamma} = 0, \tag{27}$$

where $\mathbf{n}$ is the outward-pointing unit normal vector on the boundary $\hat{\Gamma}$.

Therefore, we obtain

$$\int\limits_{\hat{\Omega}} \nabla_{\boldsymbol{x}} R_i \cdot \nabla_{\boldsymbol{x}} \xi \, d\hat{\Omega} = 0. \tag{28}$$

Similarly, from $\Delta \eta = 0$, we obtain

$$\int\limits_{\hat{\Omega}} \nabla_{\boldsymbol{x}} R_i \cdot \nabla_{\boldsymbol{x}} \eta \, d\hat{\Omega} = 0. \tag{29}$$

Finally, the equivalent variational problem of (9) in the Sobolev space H^1 reads

$$\forall R_i \in \Sigma_0 : \begin{cases} \mathbf{F}^x_{H^1} = \mathbf{0}, \\ \mathbf{F}^y_{H^1} = \mathbf{0}, \end{cases} \quad \text{s.t. } \boldsymbol{x}|_{\partial\hat{\Omega}} = \partial\Omega, \tag{30}$$

where

$$\begin{aligned}\mathbf{F}^x_{H^1} &= \int_{\hat{\Omega}} \nabla_{\boldsymbol{x}}\mathbf{R}\cdot\nabla_{\boldsymbol{x}}\xi \, d\hat{\Omega},\\ \mathbf{F}^y_{H^1} &= \int_{\hat{\Omega}} \nabla_{\boldsymbol{x}}\mathbf{R}\cdot\nabla_{\boldsymbol{x}}\eta \, d\hat{\Omega},\end{aligned} \tag{31}$$

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

By solving (30), the resulting parameterization is shown in the right of Fig. 2. One can see that the parameterization quality is significantly improved both in terms of orthogonality and uniformity and the non-uniform elements disappear. Most importantly, there is no increment of the cardinality of control points.

4.5. Analytical Jacobian computation for the discretization in H^1 space

We denote by $\partial_{\mathbf{P}^x_j}$ and $\partial_{\mathbf{P}^y_j}$ the partial derivatives w.r.t. the x-component and the y-component of the j-th control point $\mathbf{P}_j$, respectively.

Taking the partial derivatives of the inverse of the Jacobian matrix $\mathcal{J}$ gives

$$\partial_{\mathbf{P}^x_j}\mathcal{J}^{-1} = -\mathcal{J}^{-1}\begin{bmatrix}\frac{\partial R_j}{\partial\xi}, 0\\ \frac{\partial R_j}{\partial\eta}, 0\end{bmatrix}\mathcal{J}^{-1}, \tag{32}$$

and

$$\partial_{\mathbf{P}^y_j}\mathcal{J}^{-1} = -\mathcal{J}^{-1}\begin{bmatrix}0, \frac{\partial R_j}{\partial\xi}\\ 0, \frac{\partial R_j}{\partial\eta}\end{bmatrix}\mathcal{J}^{-1}. \tag{33}$$

Since $\nabla_{\boldsymbol{x}}\xi$ and $\nabla_{\boldsymbol{x}}\eta$ are merely the first and the second columns of $\mathcal{J}^{-1}$, i.e.,

$$[\nabla_{\boldsymbol{x}}\xi, \nabla_{\boldsymbol{x}}\eta] = \begin{bmatrix}\frac{\partial\xi}{\partial x} & \frac{\partial\eta}{\partial x}\\ \frac{\partial\xi}{\partial y} & \frac{\partial\eta}{\partial y}\end{bmatrix} = \mathcal{J}^{-1}, \tag{34}$$

we have

$$\begin{aligned}\partial_{\mathbf{P}^x_j}\nabla_{\boldsymbol{x}}\xi &= \left(\partial_{\mathbf{P}^x_j}\mathcal{J}^{-1}\right)_{(:,1)}, \quad \partial_{\mathbf{P}^y_j}\nabla_{\boldsymbol{x}}\xi &= \left(\partial_{\mathbf{P}^y_j}\mathcal{J}^{-1}\right)_{(:,1)},\\ \partial_{\mathbf{P}^x_j}\nabla_{\boldsymbol{x}}\eta &= \left(\partial_{\mathbf{P}^x_j}\mathcal{J}^{-1}\right)_{(:,2)}, \quad \partial_{\mathbf{P}^y_j}\nabla_{\boldsymbol{x}}\eta &= \left(\partial_{\mathbf{P}^y_j}\mathcal{J}^{-1}\right)_{(:,2)},\end{aligned} \tag{35}$$

where $(\ )_{(:,i)}$ denotes the i-th $(i = 1, 2)$ column of the matrix inside $(\ )$.

Therefore, the full Jacobian matrix for the nonlinear system (30) is

$$jac(\mathbf{F}_{H^1}) = \begin{bmatrix}\partial_{\mathbf{P}^x}\mathbf{F}^x_{H^1} & \partial_{\mathbf{P}^y}\mathbf{F}^x_{H^1}\\ \partial_{\mathbf{P}^x}\mathbf{F}^y_{H^1} & \partial_{\mathbf{P}^y}\mathbf{F}^y_{H^1}\end{bmatrix}, \tag{36}$$

where

$$\begin{aligned}(\partial_{\mathbf{P}^x}\mathbf{F}^x_{H^1})_{(i,j)} &= \int_{\hat{\Omega}} \partial_{\mathbf{P}^x_j}\mathcal{J}^{-1}\nabla_{\boldsymbol{\xi}}R_i\cdot\nabla_{\boldsymbol{x}}\xi + \nabla_{\boldsymbol{x}}R_i\cdot\partial_{\mathbf{P}^x_j}\nabla_{\boldsymbol{x}}\xi \, d\hat{\Omega},\\ (\partial_{\mathbf{P}^y}\mathbf{F}^x_{H^1})_{(i,j)} &= \int_{\hat{\Omega}} \partial_{\mathbf{P}^y_j}\mathcal{J}^{-1}\nabla_{\boldsymbol{\xi}}R_i\cdot\nabla_{\boldsymbol{x}}\xi + \nabla_{\boldsymbol{x}}R_i\cdot\partial_{\mathbf{P}^y_j}\nabla_{\boldsymbol{x}}\xi \, d\hat{\Omega},\\ (\partial_{\mathbf{P}^x}\mathbf{F}^y_{H^1})_{(i,j)} &= \int_{\hat{\Omega}} \partial_{\mathbf{P}^x_j}\mathcal{J}^{-1}\nabla_{\boldsymbol{\xi}}R_i\cdot\nabla_{\boldsymbol{x}}\eta + \nabla_{\boldsymbol{x}}R_i\cdot\partial_{\mathbf{P}^x_j}\nabla_{\boldsymbol{x}}\eta \, d\hat{\Omega},\\ (\partial_{\mathbf{P}^y}\mathbf{F}^y_{H^1})_{(i,j)} &= \int_{\hat{\Omega}} \partial_{\mathbf{P}^y_j}\mathcal{J}^{-1}\nabla_{\boldsymbol{\xi}}R_i\cdot\nabla_{\boldsymbol{x}}\eta + \nabla_{\boldsymbol{x}}R_i\cdot\partial_{\mathbf{P}^y_j}\nabla_{\boldsymbol{x}}\eta \, d\hat{\Omega}\end{aligned} \tag{37}$$

are the entries of the Jacobian blocks of nonlinear system (30).

4.6. Equivalence between nonlinear systems and fixed-point problems

The problems in (15) and (30) both come down to the following nonlinear system

$$\mathbf{F}(\mathbf{x}) = \mathbf{0},\ \mathbf{x} \in \mathbb{R}^n,\ \mathbf{F} : \mathbb{R}^n \to \mathbb{R}^n. \tag{38}$$

The typical solvers for the above problem are Newton-type methods. For instance, the truncated Newton approach and the pseudo-time-stepping Newton approach are employed in Hinz et al. (2018a). However, calculating and updating the Jacobian matrix in each iteration is mandatory in these solvers, which can be computationally expensive, particularly for large-scale problems. To circumvent this issue and to reduce the computational overhead per iteration, we first convert the nonlinear system (38) into a fixed-point iteration problem

$$\mathbf{x} = \mathbf{x} + \mathbf{F}(\mathbf{x}). \tag{39}$$

Then, the Anderson acceleration (AA) algorithm is employed to speed up the convergence of the iteration. Unlike Picard iteration which utilizes only one previous iterate, the $AA(m)$ method proceeds by linearly recombining m previous iterates such that it approximately minimizes the linearized fixed-point residual in the least-squares sense. However, the main concern related to the above basic iterative scheme (39) is that the iterates may not converge or converge very slowly.

To further improve the numerical stability and the convergence speed of AA, in the present work, we use preconditioning strategies for the fixed-point problem. Specifically, let $\mathbf{M}_k$ be a non-singular matrix called the preconditioner at iteration k. Then we define the following preconditioned fixed-point iteration scheme

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \mathbf{M}_k^{-1}\mathbf{F}(\mathbf{x}_k), \tag{40}$$

In the next section, we will introduce various types of preconditioners $\mathbf{M}_k$.

5. Nonlinear preconditioning

In this section, we will discuss how to choose a suitable preconditioner $\mathbf{M}_k$.

5.1. Preconditioners for Anderson acceleration

Assume that $\mathbf{x}^*$ is the solution to the nonlinear system $\mathbf{F}(\mathbf{x}) = \mathbf{0}$, and $\mathbf{x}_k$ is an estimate for $\mathbf{x}^*$ at the k-th iteration such that $\|\mathbf{e}_k\| = \|\mathbf{x}^* - \mathbf{x}_k\| \ll 1$. According to Taylor's expansion theorem, we have

$$\mathbf{0} = \mathbf{F}(\mathbf{x}^*) = \mathbf{F}(\mathbf{x}_k) + jac(\mathbf{F}_k)\mathbf{e}_k + \frac{1}{2}\mathbf{E}_k^{\mathrm{T}}\mathbf{H}_k\mathbf{E}_k + \mathcal{O}(\|\mathbf{e}_k\|^3), \tag{41}$$

where $jac(\mathbf{F}_k)$ is the $n \times n$ Jacobian matrix of $\mathbf{F}(\mathbf{x})$ at iteration k, $\mathbf{E}_k = diag\{\mathbf{e}_k^{\mathrm{T}}, \dots, \mathbf{e}_k^{\mathrm{T}}\}$ is a diagonal matrix composed of the transposed residual vector $\mathbf{e}_k^{\mathrm{T}}$, and

$$\mathbf{H}_k = diag\{\mathbf{H}^{(1)}(\mathbf{x}), \mathbf{H}^{(2)}(\mathbf{x}), \dots, \mathbf{H}^{(n)}(\mathbf{x})\}$$

is a block-diagonal matrix composed of $\mathbf{H}^{(i)}(\mathbf{x})$ which is the Hessian matrix of the i-th nonlinear equation $\mathbf{F}^{(i)}(x)$ $(i = 1, 2, \dots, n)$ at iteration k.

Recalling the preconditioned fixed-point iteration (40), we have

$$\mathbf{F}(\mathbf{x}_k) = \mathbf{M}_k\left(\mathbf{x}_k - \mathbf{x}_{k+1}\right) = \mathbf{M}_k\left((\mathbf{x}^* - \mathbf{x}_{k+1}) - (\mathbf{x}^* - \mathbf{x}_k)\right) = \mathbf{M}_k\left(\mathbf{e}_{k+1} - \mathbf{e}_k\right). \tag{42}$$

Inserting the above formula (42) into (41) yields

$$\begin{aligned}\mathbf{0} &= \mathbf{F}(\mathbf{x}_k) + jac(\mathbf{F}_k)\mathbf{e}_k + \frac{1}{2}\mathbf{E}_k^T\mathbf{H}_k\mathbf{E}_k + \mathcal{O}(\|\mathbf{e}_k\|^3) \\ &= \mathbf{M}_k\left(\mathbf{e}_{k+1} - \mathbf{e}_k\right) + jac(\mathbf{F}_k)\mathbf{e}_k + \frac{1}{2}\mathbf{E}_k^T\mathbf{H}_k\mathbf{E}_k + \mathcal{O}(\|\mathbf{e}_k\|^3).\end{aligned} \tag{43}$$

Multiplying $\mathbf{M}_k^{-1}$ on both sides of equation (43), we obtain

$$\begin{aligned}\mathbf{0} &= \left(\mathbf{e}_{k+1} - \mathbf{e}_k\right) + \mathbf{M}_k^{-1}jac(\mathbf{F}_k)\mathbf{e}_k + \frac{1}{2}\mathbf{M}_k^{-1}\mathbf{E}_k^T\mathbf{H}_k\mathbf{E}_k + \mathcal{O}(\|\mathbf{e}_k\|^3) \\ &= \mathbf{e}_{k+1} - \left(\mathbf{I} - \mathbf{M}_k^{-1}jac(\mathbf{F}_k)\right)\mathbf{e}_k + \frac{1}{2}\mathbf{M}_k^{-1}\mathbf{E}_k^T\mathbf{H}_k\mathbf{E}_k + \mathcal{O}(\|\mathbf{e}_k\|^3),\end{aligned} \tag{44}$$

where $\mathbf{I}$ is the $n \times n$ identity matrix.

Finally, we arrive at

$$\mathbf{e}_{k+1} = \left(\mathbf{I} - \mathbf{M}_k^{-1} jac(\mathbf{F}_k)\right)\mathbf{e}_k - \frac{1}{2}\mathbf{M}_k^{-1}\mathbf{E}_k^T\mathbf{H}_k\mathbf{E}_k + \mathcal{O}(\|\mathbf{e}_k\|^3). \tag{45}$$

If we set the preconditioner

$$\mathbf{M}_k = jac(\mathbf{F}_k) \tag{46}$$

then (45) boils down to

$$\mathbf{e}_{k+1} = -\frac{1}{2} jac(\mathbf{F}_k)^{-1}\mathbf{E}_k^T\mathbf{H}_k\mathbf{E}_k + \mathcal{O}(\|\mathbf{e}_k\|^3), \tag{47}$$

that is, $\|\mathbf{e}_{k+1}\| = \mathcal{O}(\|\mathbf{e}_k\|^2)$.

This means that, if we start from a sufficiently good initial guess and update the preconditioner $\mathbf{M}_k$ every iteration, our preconditioned AA scheme (40) may have a quadratic convergence rate, which is consistent with the standard Newton iteration. However, we run into the same issue as Newton-type solvers. The high-overhead computation of frequently updating full Jacobian is needed, contrary to our original intention for using AA. To this end, we propose several more practical preconditioning strategies to solve the involved nonlinear systems effectively.

To find a better preconditioner $\mathbf{M}_k$ for our preconditioned AA scheme (40), we consider the following preconditioning strategies.

1) **Constant preconditioner**, i.e., $\mathbf{M}_k = \alpha\mathbf{I}$, where α is a small positive real number in $(0, 1]$. This preconditioning strategy is quite simple and computationally cheapest since there is no need to compute the Jacobian matrix of the nonlinear system. However, one needs to manually tune the value of α. If one chooses $\alpha = 1$, it degenerates to the no preconditioning case. Notice that, according to (45), if the preconditioners $\mathbf{M}_k$ are too far away from the Jacobian, it only converges linearly or even diverges in some cases.
2) **Full Jacobian preconditioner**, i.e., $\mathbf{M}_k = jac(\mathbf{F}_k)$. Essentially, this preconditioning strategy is using AA to accelerate Newton's method. However, it is computationally expensive since we need to update this preconditioner at each iteration.
3) **Diagonal Jacobian preconditioner**, i.e., $\mathbf{M}_k = diag(jac(\mathbf{F}_k))$. The constant preconditioner is simple but does not necessarily work well, while the full Jacobian preconditioner is the best choice in the sense of convergence speed but is computationally expensive. To balance a "good preconditioner" and "computation overhead", this diagonal Jacobian preconditioner can be a candidate thanks to its relatively cheap computation.
4) **Block-diagonal Jacobian preconditioner**. For example, to the Jacobian matrix in (21), we adopt its diagonal blocks

$$diagBlock\,(jac(\mathbf{F})) = \begin{bmatrix} \partial_{\mathbf{P}^x}\mathbf{F}^x & \mathbf{0} \\ \mathbf{0} & \partial_{\mathbf{P}^y}\mathbf{F}^y \end{bmatrix}$$

as a preconditioner. This preconditioning strategy is a good balance between the convergence property and computation overhead. Compared with the full Jacobian preconditioner, it reduces the computational costs. It has relatively more computational costs than the diagonal Jacobian preconditioner, but it is a better approximation to the full Jacobian matrix.

Another important observation is that, unlike Newton-type solvers and gradient-based methods, the new iteration is a linear combination of several previous iterates in the AA iteration scheme. One benefit arises that frequent updating of computationally expensive preconditioners can be avoided. To enhance the computational efficiency, we update the preconditioners every N_{update} step(s) instead of updating them each iteration. In our parameterization problems, the proposed preconditioned AA scheme typically converges within 1-2 update(s) of the preconditioner.

5.2. Overview of the proposed algorithm

Our preconditioned Anderson acceleration scheme with dynamic preconditioning strategy is summarized in Algorithm 1.

Algorithm 1: Preconditioned Anderson acceleration: **PreAA(m)**.

```
Input:  $\mathbf{P}_0$: Initial guess for inner control points;
        $\mathbf{F}(\mathbf{P})$: Nonlinear system;
        $m$: Window size in AA;
        $N_{max}$: Maximum iterations;
        $tol$: Tolerance of residual norm for convergence test;
        $N_{update}$: Update the preconditioner every $N_{update}$ steps.
Output: $\mathbf{P}_{k+1}$: Optimized inner control points.
for $k = 0$; $k \le N_{max}$; $++k$ do
    Compute residual vector $\mathbf{F}_k = \mathbf{F}(\mathbf{P}_k)$;          // (15) or (30)
    // Check the termination criteria
    if $\|\mathbf{F}_k\| < tol$ then
        break;
    end
    // Update preconditioner
    if $k$ is evenly divisible by $N_{update}$ then
        Compute preconditioning matrix $\mathbf{M}$;          // Section 5.1
    end
    // Anderson acceleration
    Compute $m_k = \min\{m, k\}$;
    Compute preconditioned residual vector $\mathcal{F}_k$ by solving $\mathbf{M}\mathcal{F}_k = -\mathbf{F}_k$;
    Determine $\boldsymbol{\alpha}^{(k)} = \left(\alpha_0^{(k)}, \cdots, \alpha_{m_k}^{(k)}\right)^T$ that solves
        $\underset{\boldsymbol{\alpha}=(\alpha_0,\cdots,\alpha_{m_k})^T}{\arg\min} \left\| \sum_{i=0}^{m_k} \mathcal{F}_{k-m_k+i}\boldsymbol{\alpha} \right\|_2^2, \quad \text{s.t.} \quad \sum_{i=0}^{m_k} \alpha_i = 1;$
    Update $\mathbf{P}_{k+1} = \sum_{i=0}^{m_k} \alpha_i^{(k)} \left(\mathbf{P}_{k-m_k+i} + \mathcal{F}_{k-m_k+i}\right)$;
end
```

6. Numerical experiments

In this section, we numerically investigate our H^1 discretized parameterization technique and compare different preconditioning strategies in detail. In addition, comparisons with state-of-the-art methods Ji et al. (2021, 2022b) are also performed.

6.1. Implementation details

The proposed method is implemented using C++. All numerical experiments are conducted on a laptop computer (MacBook Pro 14-inch 2021, Apple M1 Pro CPU, and 16-GB RAM). The present implementation is based on the open-source C++ library G+Smo (Geometry + Simulation Modules) Jüttler et al. (2014); Mantzaflaris (2019). Eigen Guennebaud et al. (2010), a widely used C++ template library for linear algebra, is employed in matrix/vector operations and for solving the involved linear systems.

Algorithm's parameter settings: there are several parameters in Algorithm 1. All involved parameters are set as default values though the convergence speed can be enhanced by tuning the involved parameters. We set the window size $m = 5$, and $N_{update} = 10$ means that the preconditioner is updated every 10 iterations. As to the stopping criteria, we set the maximum iterations $N_{max} = 1000$ and the tolerance for the residual norm $tol = 1e - 5$, which proved to work well in our experiments.

6.2. Quality metrics for parameterization

The orthogonality and uniformity of the resulting parameterization significantly affect the numerical accuracy of the subsequent analysis Pilgerstorfer and Jüttler (2014). Therefore, two quality metrics are employed here, one is for orthogonality, and the other is for uniformity.

- Scaled Jacobian

$$m_{SJ} = \frac{|\mathcal{J}|}{\|\boldsymbol{x}_\xi\| \|\boldsymbol{x}_\eta\|}. \tag{48}$$

It is easy to see that $-1 \le m_{SJ} \le 1$. For a bijective parameterization, $m_{SJ} > 0$ holds, for $\forall \boldsymbol{\xi} \in \hat{\Omega}$. Once a negative value of m_{SJ} is detected, the parameterization $\boldsymbol{x}$ is folded somewhere. It reaches its optimal value 1 if the orthogonality of $\boldsymbol{x}$ perseveres.

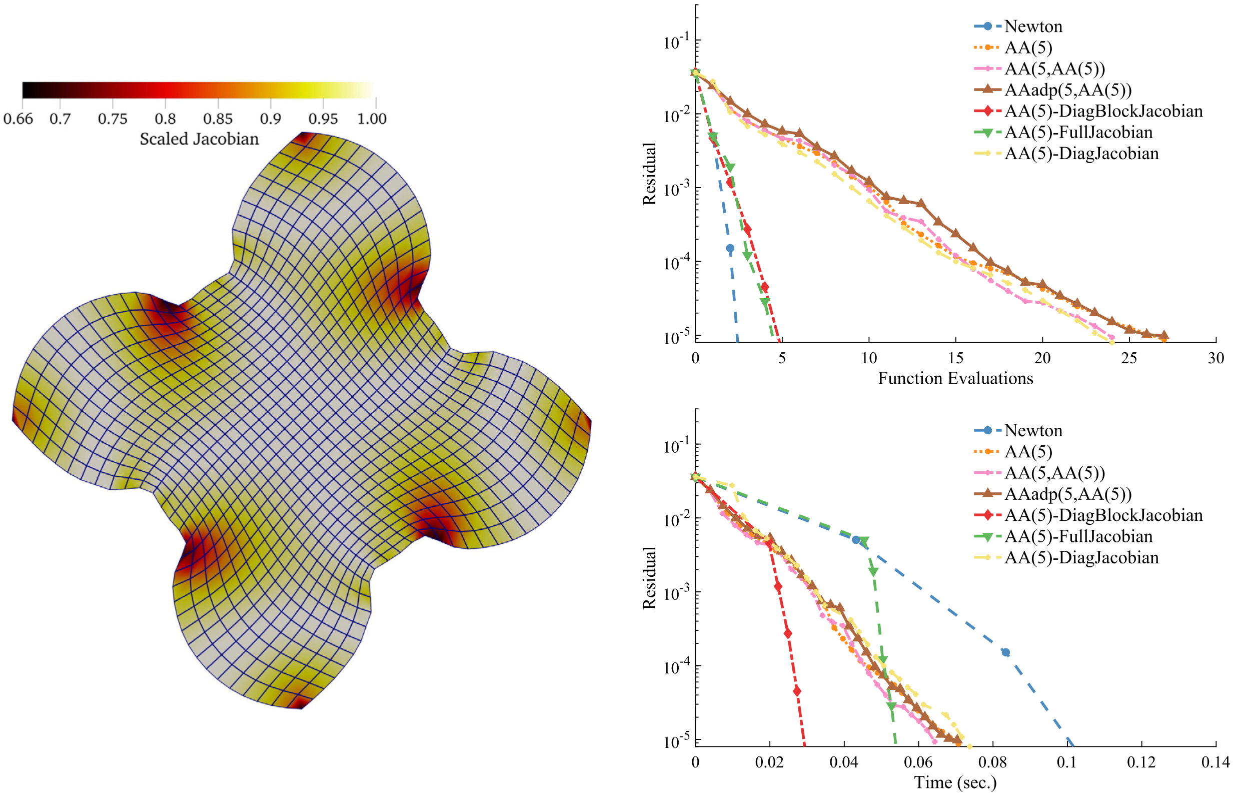


Fig. 3. Male rotor example: performance comparisons between different solvers, including standard Newton's iteration, AA(5), AA(5) composite AA(5), adaptive AA(5) composite AA(5), preconditioned AA(5) with block-diagonal Jacobian preconditioner, AA(5) with diagonal Jacobian preconditioner, and AA(5) with full Jacobian preconditioner.

- Uniformity metric

$$m_{unif.} = \left(\frac{|\mathcal{J}|}{R_{area}} - 1\right)^2, \tag{49}$$

where $R_{area} = Area(\Omega)/Area(\hat{\Omega})$ is the area ratio of the parametric domain $\hat{\Omega}$ and the computational domain Ω. This definition is based on the fact that the Jacobian determinant describes the area ratio of $\boldsymbol{x}$ at a given point. $m_{unif.}$ attains its minimum value 0 if the parameterization conserves uniformity.

In our experiments, both quality metrics are evaluated on a dense sampling of 1001 × 1001 points including the boundaries. In our statistics, the maximum values of m_{SJ} and the minimum values of $m_{unif.}$ are omitted, since they are attainable in every example.

6.3. Performance comparisons between different solvers

In this section, we study the numerical behavior of our preconditioned AA scheme with different preconditioning strategies. Comparisons with other solvers are also provided, including the standard Newton's method, the standard AA algorithm (AA(m)), and two recently developed methods Chen and Vuik (2022b): the AA composite AA method (AA(m,AA(m))) and the AA adaptive composite AA method (AAadp(m,AA(m))).

Fig. 3 shows a computational domain shaped by a male rotor. We start from its B-Rep composed of four B-spline curves and initialize the inner control points by computationally inexpensive Spring patch construction Gravesen et al. (2012). The residual norm of different solvers to function evaluations and computational time are shown in the right. One can see that Newton's method shows the best convergence speed in terms of function evaluations, which converges within 4 iterations. In this example, AA(5), AA(5, AA(5)), AAadp(5, AA(5)), and the preconditioned AA scheme with diagonal Jacobian preconditioner converge in 25 iterations. Our preconditioned AA scheme with full Jacobian preconditioner and preconditioned AA scheme with block-diagonal Jacobian preconditioner converge after 9 and 10 iterations, respectively, which is worse than the standard Newton's method but better than the other AA schemes.

In this example, however, all AA schemes beat standard Newton's method in terms of computational time. Although the analytical Jacobian matrix formulas are derived in (21), which greatly enhances the computational efficiency, the update of

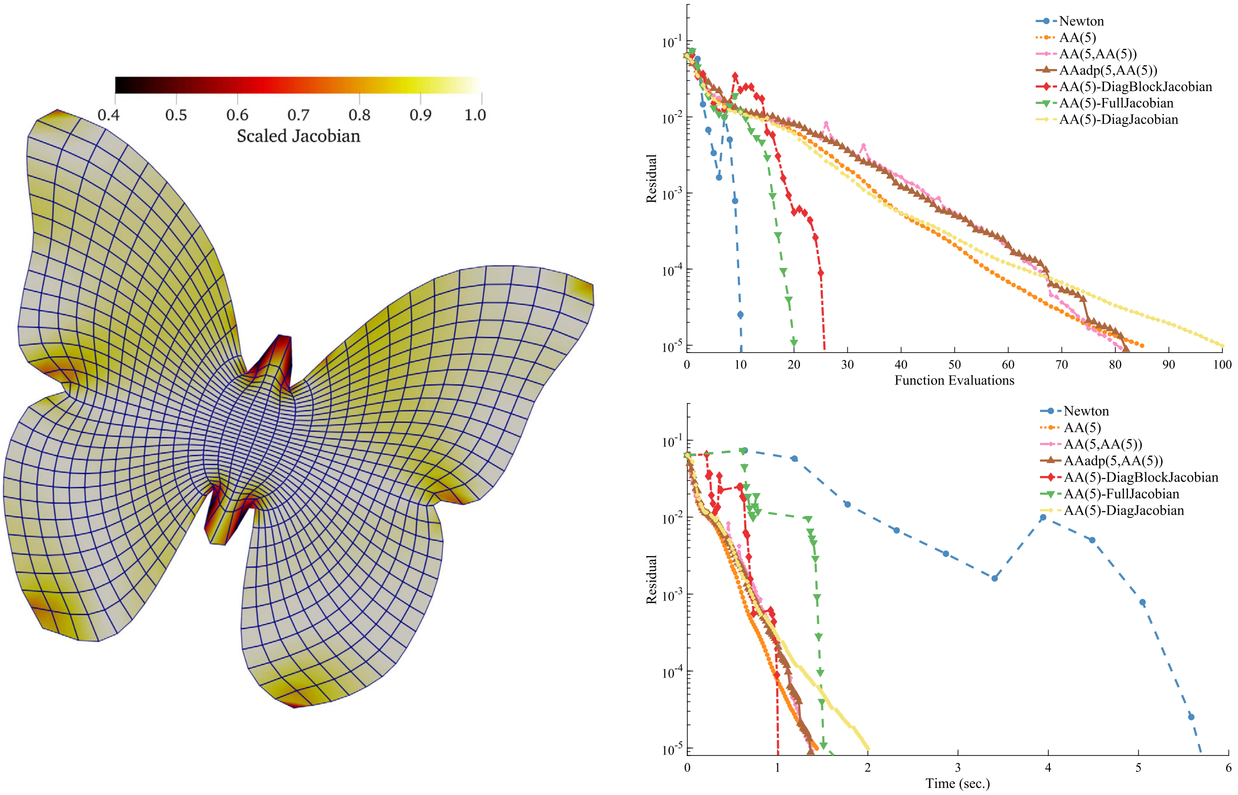


Fig. 4. Butterfly example: performance comparisons between different solvers, including standard Newton's iteration, AA(5), AA(5) composite AA(5), adaptive AA(5) composite AA(5), preconditioned AA(5) with block-diagonal Jacobian preconditioner, AA(5) with diagonal Jacobian preconditioner, and AA(5) with full Jacobian preconditioner.

the full Jacobian matrix each iteration in standard Newton's method can be still expensive. Notice that, our preconditioned AA scheme with block-diagonal Jacobian preconditioner is over four times faster than standard Newton's method.

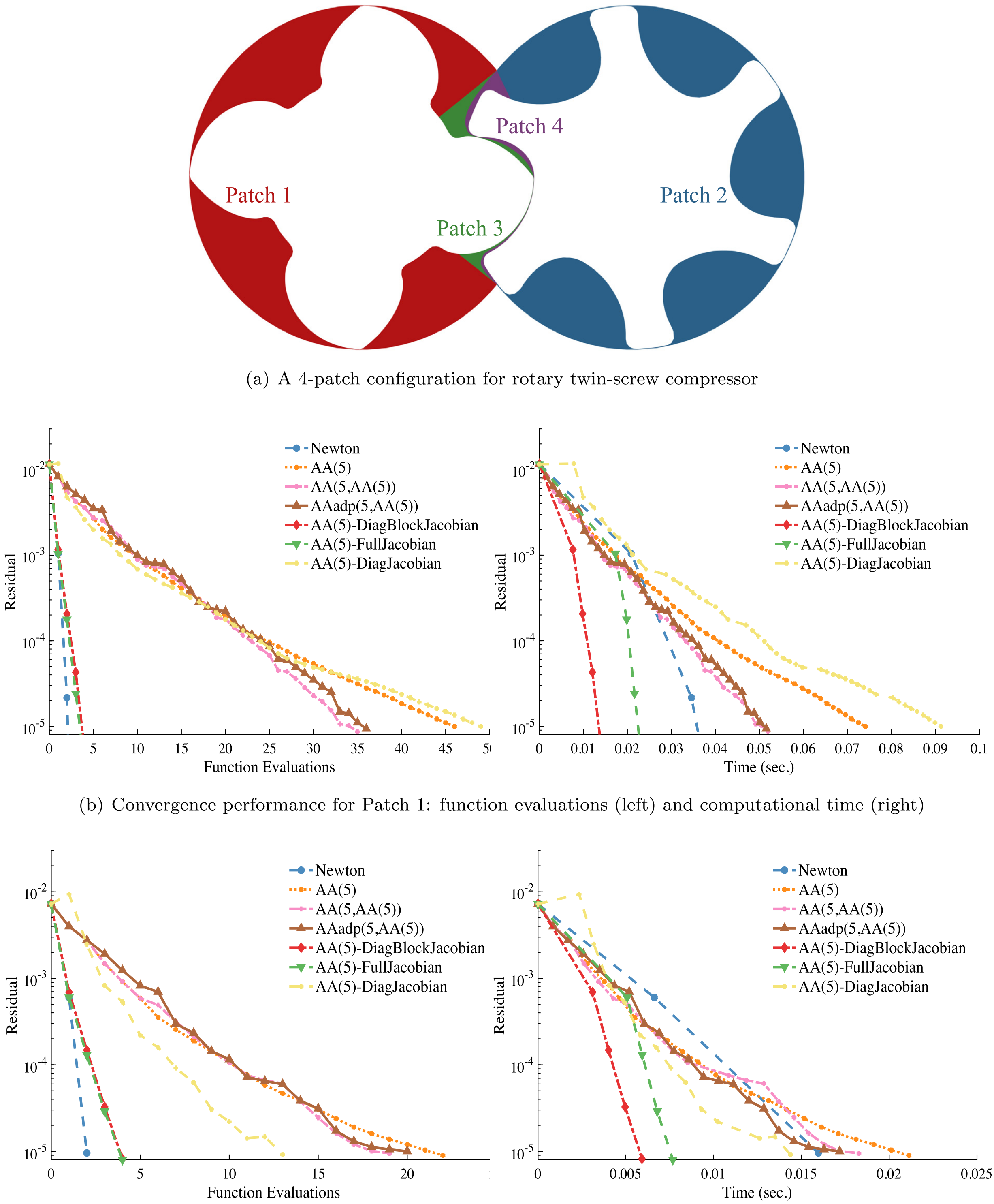
As shown in Fig. 4, a similar numerical performance can be observed for the butterfly geometry. In this example, our preconditioned AA scheme with the block-diagonal Jacobian preconditioner is almost six times faster than standard Newton's method. Here, oscillations on the convergence curves of our preconditioned AA scheme are found. This is because the original (no preconditioning) residual norms are tracked for a fair comparison, and the preconditioned residual norms still descend steadily.

Next, we conduct tests on various nonlinear solvers on a multi-patch parameterization. Fig. 5(a) displays a single slice of a rotary twin-screw compressor that is parameterized by a 4-patch B-spline. The convergence performance of different nonlinear solvers on two representative patches, namely Patch 1 and Patch 4, is shown in Fig. 5(b) and Fig. 5(c), respectively. Our preconditioned AA scheme with a block-diagonal Jacobian preconditioner performs the best in terms of computational time. The other two patches exhibit similar numerical performance.

6.4. Parameterization quality improvement by scaled H^1 discretization

As mentioned in Sec. 4.4, the parameterizations produced by the original discretized nonlinear system (15) in H^2 space may have non-uniform elements. Specifically, the resulting grid lines are attracted to the concave part and repelled from the convex part of the boundary, as shown in Fig. 2. To this end, we introduce a scaled version of harmonic equations (9) in Sobolev space H^1.

Table 1 summarizes the parameterization quality metrics of different models by the H^2 discretization and the H^1 discretization. One can see that our scaled H^1 discretization greatly improves the parameterization quality. A typical example is shown in Fig. 6. By solving the nonlinear system (15), the resulting parameterization is with folds inside the red circle, shown in the left. The negative scaled Jacobian in Table 1 also validates it. This example shows that though the bijectivity of harmonic mapping is guaranteed by Radó-Kneser-Choquet theorem in an infinite-dimensional space, it may produce invalid parameterization in a finite-dimensional spline space. To this end, Hinz et al. (2018a) suggest that a better approximation to the harmonic mapping can be obtained by several refinements. However, the problem is that refinements lead to more control points, and thus the model becomes heavier. The result produced by (30), with dramatically improved parameteriza-



(a) A 4-patch configuration for rotary twin-screw compressor

(b) Convergence performance for Patch 1: function evaluations (left) and computational time (right)

(c) Convergence performance for Patch 4: function evaluations (left) and computational time (right)

Fig. 5. Performance comparisons between different solvers over 4-patch parameterized rotary twin-screw geometry.

Table 1
Comparisons between different parameterization methods, including barrier-function based method Ji et al. (2021), penalty function-based method Ji et al. (2022b), H^2 discretization (15), and H^1 discretization (30). The first two methods are solved by a standard L-BFGS method, whereas the last two methods (namely, those in H^2 space and H^1 space) are solved using our preconditioned Anderson acceleration method with the block-diagonal Jacobian preconditioning strategy. The minimum and the average values of scaled Jacobian m_{SJ}, the maximum and the average values of the uniformity metric $m_{unif.}$, and the computational timings T_{total} (in seconds) are reported. All results with the best performance are highlighted in bold.

Model	#DOFs	Method	m_{SJ}		$m_{unif.}$		$T_{total}(s)$
			min.	avg.	max.	avg.	
Duck (Fig. 2)	160	Ji et al. (2021)	0.2986	0.9243	62.9882	0.5913	0.0393
		Ji et al. (2022b)	0.2986	0.9243	62.9875	0.5912	0.0722
		H^2 space (15)	0.2444	**0.9276**	131.3254	0.7845	**0.0066**
		H^1 space (30)	**0.3022**	0.9257	**27.0234**	**0.3545**	0.0195
Male Rotor (Fig. 3)	1352	Ji et al. (2021)	0.3737	**0.9713**	2.7579	0.1444	0.2204
		Ji et al. (2022b)	0.3738	**0.9713**	2.7611	0.1444	0.2984
		H^2 space (15)	0.3140	**0.9713**	2.7749	0.1451	**0.0590**
		H^1 space (30)	**0.5293**	0.9644	**0.7342**	**0.0688**	0.1544
Butterfly (Fig. 4)	4418	Ji et al. (2021)	0.3095	0.9414	77.1353	0.6885	16.0262
		Ji et al. (2022b)	**0.3096**	0.9414	77.1177	0.6886	27.4425
		H^2 space (15)	0.2913	0.9416	114.4743	0.7014	**1.5317**
		H^1 space (30)	0.2319	**0.9596**	**30.2282**	**0.3838**	4.5948
Plane (Fig. 6)	144	Ji et al. (2021)	0.3575	0.9145	45.6981	1.0526	**0.0385**
		Ji et al. (2022b)	0.3575	0.9145	45.6983	1.0526	0.0539
		H^2 space (15)	-1	-	-	-	-
		H^1 space (30)	**0.3716**	**0.9379**	**28.6562**	**0.7400**	0.0538
Male screw (Fig. 7)	1616	Ji et al. (2021)	-1	-	-	-	-
		Ji et al. (2022b)	-1	-	-	-	-
		H^2 space (15)	0.3469	0.9407	36.5805	4.6449	**0.0436**
		H^1 space (30)	**0.4611**	**0.9474**	**26.1502**	**4.5278**	0.0531
Female screw (Fig. 7)	1616	Ji et al. (2021)	-1	-	-	-	-
		Ji et al. (2022b)	-1	-	-	-	-
		H^2 space (15)	0.2041	0.9443	60.6725	4.9954	**0.0344**
		H^1 space (30)	**0.4388**	**0.9539**	**33.6031**	**4.7157**	0.1320

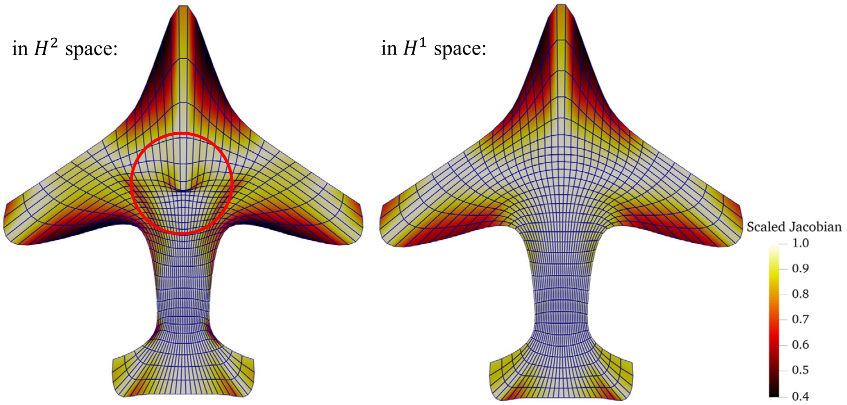


Fig. 6. Plane example: the left shows the parameterization produced by the discretization (15) in the H^2 space, where some folds can be observed inside the red circle. The right shows the result produced by the discretization (30) in the H^1 space. The color encodes the scaled Jacobian, with white representing optimal orthogonality.

tion quality, is shown in the right of Fig. 6. Our method generates high-quality parameterization results with fewer control points, which is preferable in practice.

6.5. Comparisons with state-of-the-art parameterization approaches

In this section, we compare the proposed approach with state-of-the-art parameterization approaches. Specifically, two methods are considered. One is the barrier function-based approach Ji et al. (2021). Another one is the so-called penalty function-based approach Ji et al. (2022b). Both methods attempt to approximate the harmonic mapping by minimizing

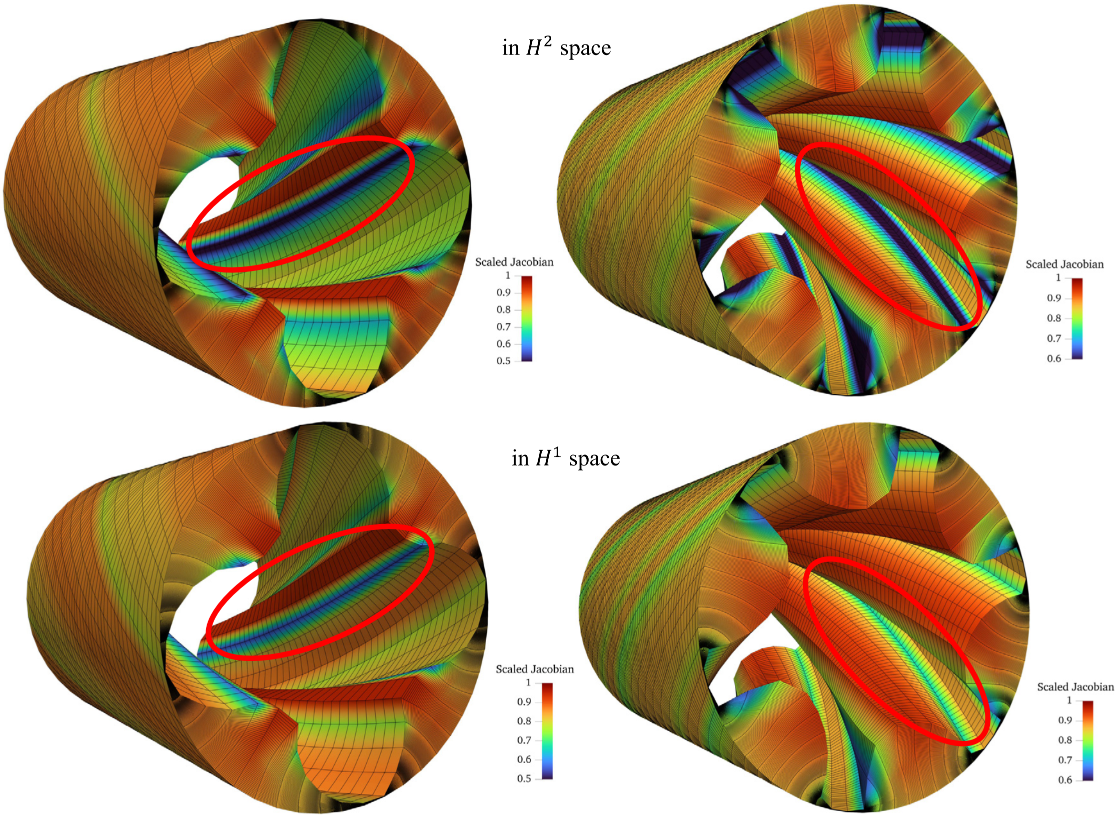


Fig. 7. Screw examples: the first row shows the parameterizations generated by the discretization (15) in the H^2 space for the male screw geometry and the female screw geometry. The second row shows the parameterizations generated by the discretization (30) in the H^1 space for the male screw geometry and the female screw geometry. The color encodes the scaled Jacobian, with red representing optimal orthogonality.

Winslow's function. Different from the barrier function-based approach, the penalty function-based approach avoids the foldover elimination steps by introducing a penalty term. Therefore, the resulting parameterizations should be the same upon convergence. Specifically, the following nonlinear optimization problem is solved in these two approaches

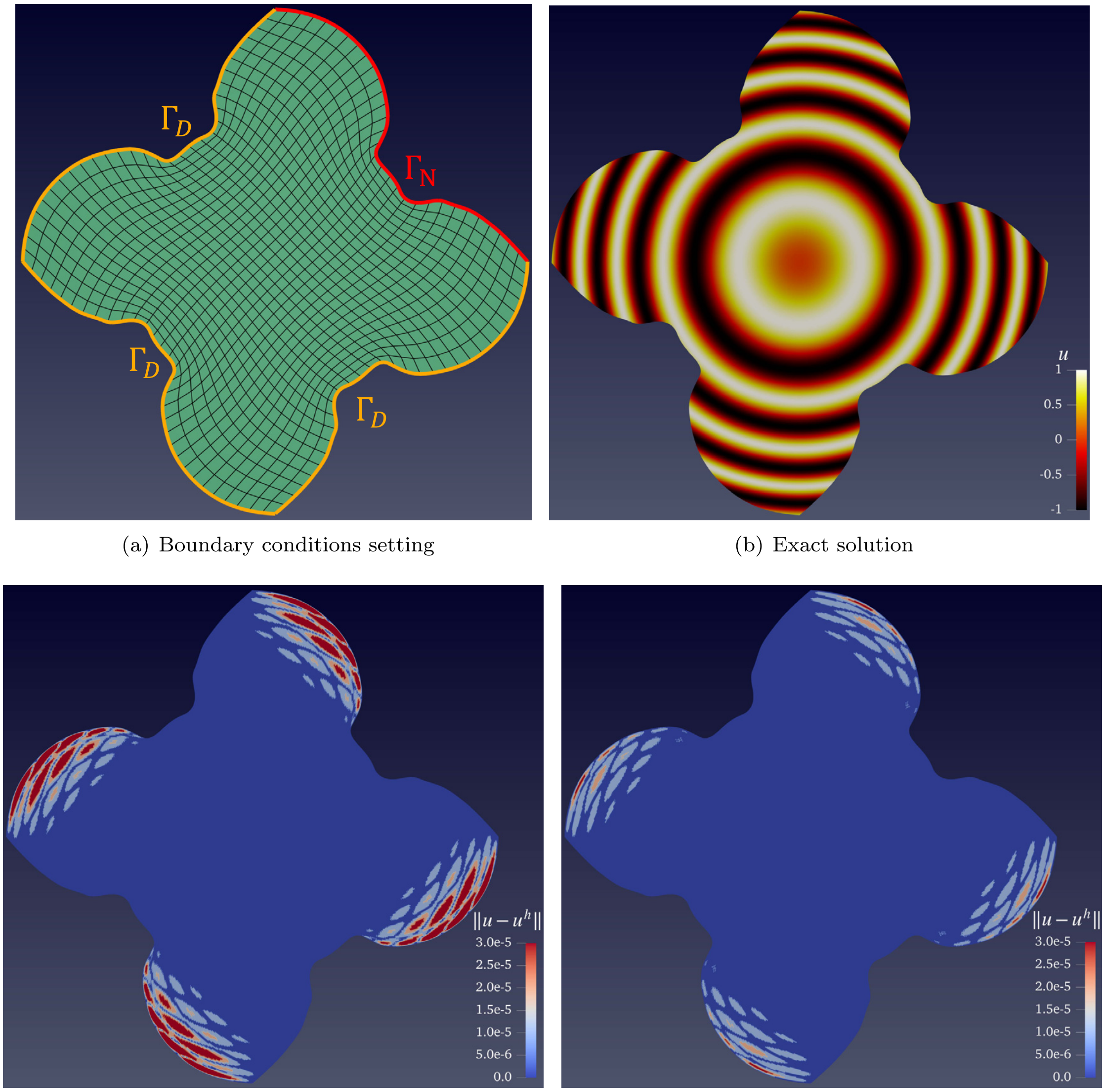
$$\underset{\mathbf{P}_i, i\in\mathcal{I}_I}{\arg\min} \int_{\hat{\Omega}} \left(E_D + \lambda E_{unif.}\right) d\hat{\Omega}, \tag{50}$$

where E_D is Winslow's function as (11). To demonstrate the difference between different discretization methods, we set the trade-off parameter λ to 0 in our numerical experiments.

For a fair comparison, we re-implement these two approaches under the same computational environment (G+Smo), which is much more efficient than the original implementation using MATLAB. Interested readers may refer to the aforementioned papers for more comparisons with the existing approaches.

The quality metrics and computational timings of different parameterization methods are reported in Table 1. As we expected, the barrier function-based method Ji et al. (2021) and the penalty function-based method Ji et al. (2022b) generate almost the same parameterization results in all models since both of them attempt to minimize Winslow's function. Our discretization in the Sobolev space H^1 generates higher-quality results in terms of not only orthogonality but also uniformity. As to computational efficiency, our method is much more efficient than the other methods except for the Plane example. Notice that, for the barrier and penalty function-based method with the Limited-memory BFGS solver, a slow convergence behavior is observed in the Butterfly example with 4418 degrees-of-freedom. However, our method is almost 4 times faster than these two methods.

As shown in Fig. 7, the resulting planar parameterizations can be adopted to generate 2.5-D volumetric parameterizations using extrude operations in some industrial applications. To complete these parameterizations, we first construct an analysis-suitable planar parameterization for a single slice, apply a canonical rotation matrix to the resulting control points, and adopt an extrude operation along the z-direction in the physical space. After the extrude operation, the resulting parameterization has a degree of 1 in the third parameter direction. To these geometries with extreme aspect ratios, the methods based on minimizing the corrected Winslow's function Ji et al. (2021, 2022b) encounter numerical difficulty, which is also observed in Fig. 1. As shown in Table 1, the minimum value of the scaled Jacobian is negative, which means that the resulting



(a) Boundary conditions setting

(b) Exact solution

(c) Absolute error for the H^2 discretization

(d) Absolute error for our H^1 discretization

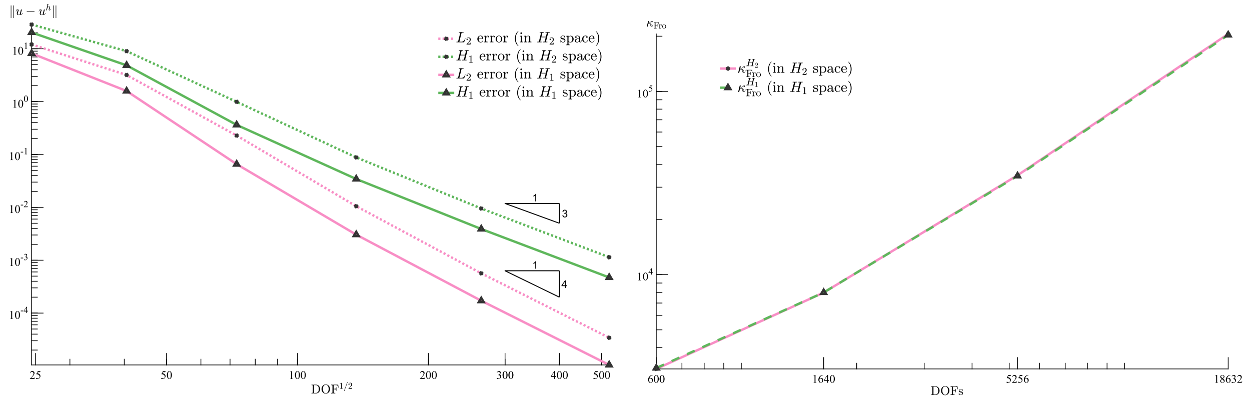
Fig. 8. Problem setting and absolute error colormap of the 2D male rotor geometry. Here, the absolute errors in (c) and (d) are computed with DOFs = 69960.

parameterizations are invalid. The discretization in H^2 and H^1 space produce valid parameterizations. It can be seen in Fig. 7 that the discretization in H^2 space produces higher quality results. As a downside, the discretization in H^2 space needs more computational time. However, by using the proposed Anderson acceleration with the block-diagonal Jacobian preconditioning strategy, its computational timing is still acceptable as shown in Table 1.

7. Application to IGA simulation

To demonstrate that our resulting parameterizations are applicable for IGA, we study the following Poisson's problem with mixed boundary conditions over physical domain Ω:

$$\begin{cases} -\Delta u = f, & \text{in } \Omega, \\ u = g, & \text{on } \Gamma_D, \\ \mathbf{n} \cdot \nabla u = h, & \text{on } \Gamma_N, \end{cases} \tag{51}$$



(a) Error convergence history for h-refinement (b) Frobenius condition number history of stiffness matrix

Fig. 9. Error convergence and Frobenius condition number of stiffness matrix during h-refinement.

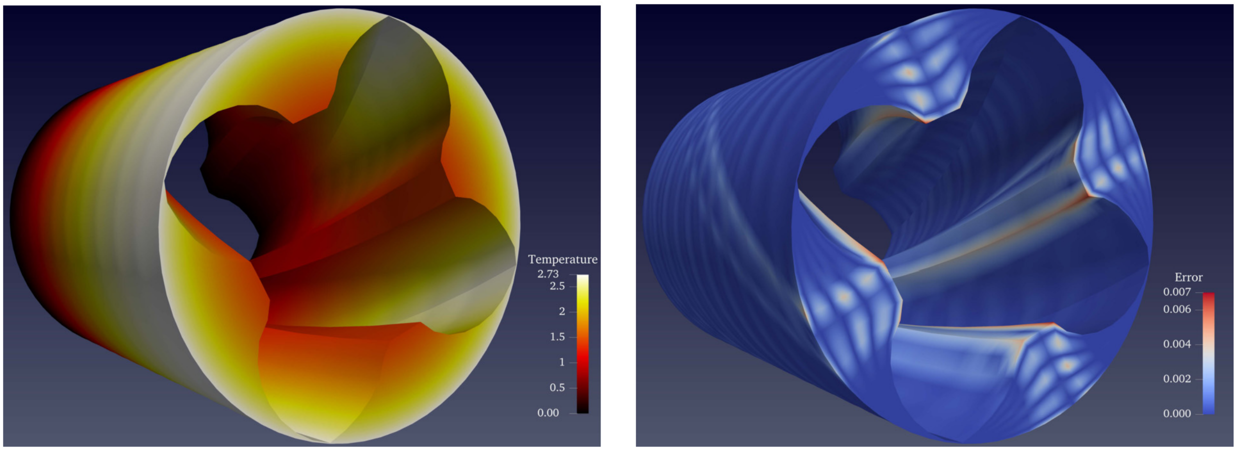


Fig. 10. IGA solution and absolute error colormap of 3D screw geometry (DOFs = 30704).

where $f \in L^2(\Omega) : \Omega \to \mathbb{R}$ is a given source term, $\mathbf{n}$ is the outward pointing unit normal vector, $\partial\Omega = \bar{\Gamma}_D \cup \bar{\Gamma}_N$ (with $\Gamma_D \cap \Gamma_N = \varnothing$) defines the boundary, Γ_D and Γ_N are the parts of the boundary where Dirichlet and Neumann boundary conditions are prescribed, respectively.

To begin with, we examine Poisson's equation (51) over the male rotor depicted in Fig. 8(a). In this scenario, mixed boundary conditions are applied to the boundaries, and Dirichlet boundary conditions are imposed using L_2 projection. The exact solution, shown in Fig. 8(b), is given by $u^{\text{exact}} = \sin(\frac{x^2+y^2}{50})$, while the source term is $f = -\Delta u^{\text{exact}}$. As demonstrated in Fig. 8(c) and Fig. 8(d), different parameterizations employing the same bi-cubic B-splines exhibit varying levels of absolute error. Notably, the parameterization constructed by the H^1 discretization yields a lower error level than that constructed by the H^2 discretization.

Fig. 9(a) illustrates the error convergence history during h-refinement. The plot indicates that our parameterization is IGA-suitable, as both L^2 norm error and H^1 norm error achieve the theoretical optimal convergence rate. Fig. 9(b) displays the Frobenius condition number history of the stiffness matrices during h-refinement. Consistent with observations in Ji et al. (2022a), the two resulting parameterizations exhibit a steady, linear increase in the stiffness matrix condition number as the number of degrees-of-freedom increases, while maintaining a low level (approximately 10^5 with DOFs=18632). It indicates that our parameterization method is reliable and suitable for practical applications in IGA.

Next, we solve Poisson's equation (51) over the volumetric male screw geometry shown in Fig. 7. Prior to the IGA simulation, we perform one degree elevation along the z-direction to obtain a tri-quadratic B-spline representation. The exact solution is $u^{\text{exact}} = e^{\frac{x^2+y^2}{r^2}} \sin(\frac{z\pi}{m})$, where $r = 36$ and $m = 200$.

The numerical solution and its corresponding absolute error colormap, with degrees-of-freedom (DOFs) of 30704, are shown in Fig. 10. In Fig. 11, we plot the error convergence results for uniform h-refinements. One can see that the theoretical optimal convergence rate is attained for both L^2 norm error and H^1 norm error, which implies that our parameterization is IGA-suitable.

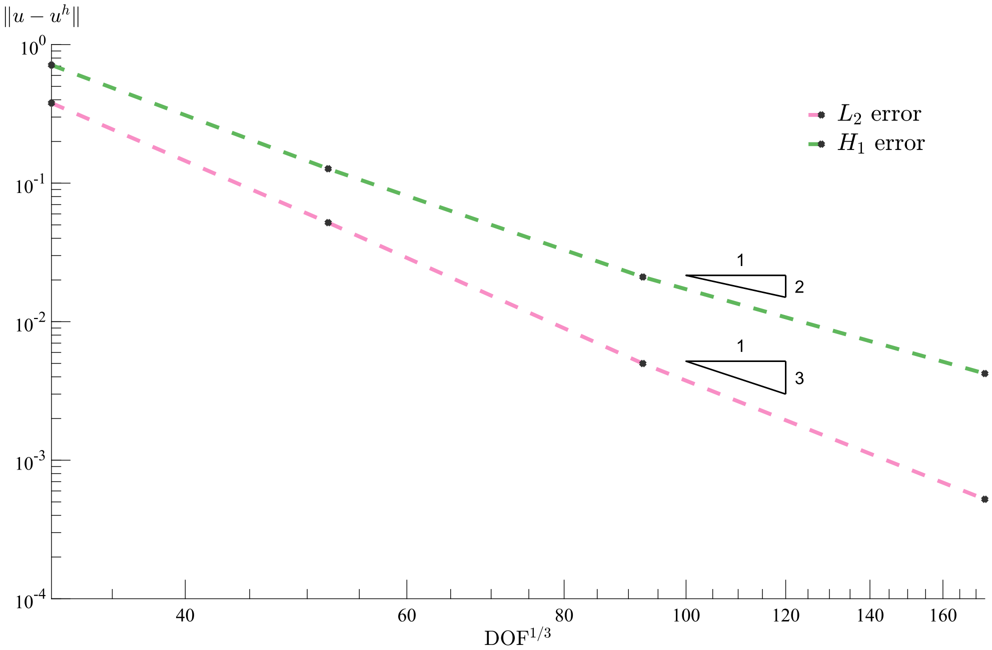


Fig. 11. Error convergence history for h-refinement.

8. Conclusions and outlook

In this paper, we propose a novel preconditioned Anderson acceleration framework to enhance the numerical stability and computational efficiency of the PDE-based elliptic parameterization technique. The discretization of the scaled harmonic mapping is developed in the Sobolev space H^1 to alleviate the inherent issue of the elliptic parameterization method, i.e., non-uniform elements near the concave and convex boundaries. The Jacobian matrices of the involved nonlinear systems are derived analytically, which greatly improves the computational efficiency. Numerical experimental results show that our preconditioned AA scheme effectively reduces the computational overhead per iteration and the total computational time. Compared with state-of-the-art approaches, our method shows better numerical performance and adaptation to challenging geometries with extreme aspect ratios.

In our parameterization problem, the analytical Jacobian matrices for the related nonlinear systems are available by careful derivation. However, for many other problems, an analytical Jacobian matrix may be difficult or even impossible to obtain. Therefore, a Jacobian-free preconditioner is worthy for investigation.

Multi-patch parameterizations are essential for complex geometries, particularly those with high genus topology. Using domain partitioning techniques to construct high-quality, analysis-suitable parameterizations for general CAD models from their boundary representations is one of our ongoing works. Besides, extending our elliptic parameterization approach to the fully volumetric case is of vital importance to practical applications. Here, one main concern is the lack of a theoretical bijectivity guarantee in 3D.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgements

We thank all the anonymous reviewers for their valuable comments and constructive suggestions. This research was partially funded by the National Natural Science Foundation of China (Grant No. 12001287), and the Startup Foundation for Introducing Talent of Nanjing University of Information Science and Technology (Grant No. 2019r106). Additionally, Ye Ji and Kewang Chen gratefully acknowledge the partial support provided by the China Scholarship Council (Grant Nos. 202106060082 and 202008320191) during their visit to Delft University of Technology.

References

Anderson, D.G., 1965. Iterative procedures for nonlinear integral equations. J. ACM 12, 547–560.

Anderson, D.G., 2019. Comments on "Anderson acceleration, mixing and extrapolation". Numer. Algorithms 80, 135–234.

Bastl, B., Slabá, K., 2021. Planar multi-patch domain parameterization for isogeometric analysis based on evolution of fat skeleton. Comput. Methods Appl. Mech. Eng. 385, 114045.

Bian, W., Chen, X., Kelley, C., 2021. Anderson acceleration for a class of nonsmooth fixed-point problems. SIAM J. Sci. Comput. 43, S1–S20.

Brune, P.R., Knepley, M.G., Smith, B.F., Tu, X., 2015. Composing scalable nonlinear algebraic solvers. SIAM Rev. 57, 535–565.

Buchegger, F., Jüttler, B., 2017. Planar multi-patch domain parameterization via patch adjacency graphs. Comput. Aided Des. 82, 2–12.

Carlson, N.N., Miller, K., 1998. Design and application of a gradient-weighted moving finite element code I: in one dimension. SIAM J. Sci. Comput. 19, 728–765.

Chan, C.L., Anitescu, C., Rabczuk, T., 2017. Volumetric parametrization from a level set boundary representation with PHT-splines. Comput. Aided Des. 82, 29–41.

Chen, K., Vuik, C., 2022a. Composite Anderson acceleration method with two window sizes and optimized damping. Int. J. Numer. Methods Eng. 123, 5964–5985.

Chen, K., Vuik, C., 2022b. Non-stationary Anderson acceleration with optimized damping. ArXiv preprint. arXiv:2202.05295.

Chen, L., Bu, N., Jin, Y., Xu, G., Li, B., 2022. Construction of iga-suitable volume parametric models by the segmentation–mapping–merging mechanism of design features. Comput. Aided Des. 146, 103228.

Chen, L., Xu, G., Wang, S., Shi, Z., Huang, J., 2019. Constructing volumetric parameterization based on directed graph simplification of l_1 polycube structure from complex shapes. Comput. Methods Appl. Mech. Eng. 351, 422–440.

Cohen, E., Martin, T., Kirby, R., Lyche, T., Riesenfeld, R., 2010. Analysis-aware modeling: understanding quality considerations in modeling for isogeometric analysis. Comput. Methods Appl. Mech. Eng. 199, 334–356.

Duren, P., Hengartner, W., 1997. Harmonic mappings of multiply connected domains. Pac. J. Math. 180, 201–220.

Evans, C., Pollock, S., Rebholz, L.G., Xiao, M., 2020. A proof that Anderson acceleration improves the convergence rate in linearly converging fixed-point methods (but not in those converging quadratically). SIAM J. Numer. Anal. 58, 788–810.

Falini, A., Jüttler, B., 2019. THB-splines multi-patch parameterization for multiply-connected planar domains via template segmentation. J. Comput. Appl. Math. 349, 390–402.

Falini, A., Špeh, J., Jüttler, B., 2015. Planar domain parameterization with THB-splines. Comput. Aided Geom. Des. 35, 95–108.

Fang, H.r., Saad, Y., 2009. Two classes of multisecant methods for nonlinear acceleration. Numer. Linear Algebra Appl. 16, 197–221.

Farin, G., Hansford, D., 1999. Discrete coons patches. Comput. Aided Geom. Des. 16, 691–700.

Garanzha, V., Kaporin, I., 1999. Regularization of the barrier variational method. Comput. Math. Math. Phys. 39, 1426–1440.

Garanzha, V., Kaporin, I., Kudryavtseva, L., Protais, F., Ray, N., Sokolov, D., 2021. Foldover-free maps in 50 lines of code. ACM Trans. Graph. 40, 1–16.

Gravesen, J., Evgrafov, A., Nguyen, D.M., Nørtoft, P., 2012. Planar parametrization in isogeometric analysis. In: International Conference on Mathematical Methods for Curves and Surfaces. Springer, pp. 189–212.

Guennebaud, G., Jacob, B., et al., 2010. Eigen v3. http://eigen.tuxfamily.org.

Haberleitner, M., Jüttler, B., Masson, Y., 2019. Isogeometric segmentation via midpoint subdivision suitable solids. Comput. Aided Des. 114, 179–190.

Haelterman, R., Degroote, J., Van Heule, D., Vierendeels, J., 2010. On the similarities between the quasi-Newton inverse least squares method and GMRES. SIAM J. Numer. Anal. 47, 4660–4679.

Hinz, J., 2020. PDE-Based Parameterization Techniques for Isogeometric Analysis Applications. Ph.D. thesis. Delft University of Technology.

Hinz, J., Möller, M., Vuik, C., 2018a. Elliptic grid generation techniques in the framework of isogeometric analysis applications. Comput. Aided Geom. Des. 65, 48–75.

Hinz, J., Möller, M., Vuik, C., 2018b. Spline-based parameterization techniques for twin-screw machine geometries. In: IOP Conference Series: Materials Science and Engineering. IOP Publishing, 012030.

Hormann, K., Greiner, G., 2000. MIPS: an efficient global parametrization method. Technical Report. Erlangen-Nuernberg Univ., Germany, Computer Graphics Group.

Hughes, T.J., Cottrell, J.A., Bazilevs, Y., 2005. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. Comput. Methods Appl. Mech. Eng. 194, 4135–4195.

Ji, Y., Li, J.G., Yu, Y.Y., Zhu, C.G., 2022a. h-Refinement method for toric parameterization of planar multi-sided computational domain in isogeometric analysis. Comput. Aided Geom. Des. 93, 102065.

Ji, Y., Wang, M.Y., Pan, M.D., Zhang, Y., Zhu, C.G., 2022b. Penalty function-based volumetric parameterization method for isogeometric analysis. Comput. Aided Geom. Des. 94, 102081.

Ji, Y., Wang, M.Y., Wang, Y., Zhu, C.G., 2022c. Curvature-based r-adaptive planar NURBS parameterization method for isogeometric analysis using bi-level approach. Comput. Aided Des. 150, 103305.

Ji, Y., Yu, Y.Y., Wang, M.Y., Zhu, C.G., 2021. Constructing high-quality planar NURBS parameterization for isogeometric analysis by adjustment control points and weights. J. Comput. Appl. Math. 396, 113615.

Jüttler, B., Langer, U., Mantzaflaris, A., Moore, S.E., Zulehner, W., 2014. Geometry + simulation modules: implementing isogeometric analysis. PAMM 14, 961–962.

Lin, L., Yang, C., 2013. Elliptic preconditioner for accelerating the self-consistent field iteration in Kohn–Sham density functional theory. SIAM J. Sci. Comput. 35, S277–S298.

Liu, H., Yang, Y., Liu, Y., Fu, X.M., 2020. Simultaneous interior and boundary optimization of volumetric domain parameterizations for IGA. Comput. Aided Geom. Des. 79, 101853.

Mantzaflaris, A., 2019. An overview of geometry plus simulation modules. In: International Conference on Mathematical Aspects of Computer and Information Sciences. Springer, pp. 453–456.

Martin, T., Cohen, E., Kirby, R.M., 2009. Volumetric parameterization and trivariate B-spline fitting using harmonic functions. Comput. Aided Geom. Des. 26, 648–664.

Nguyen, T., Jüttler, B., 2010. Parameterization of contractible domains using sequences of harmonic maps. In: International Conference on Curves and Surfaces. Springer, pp. 501–514.

Nian, X., Chen, F., 2016. Planar domain parameterization for isogeometric analysis based on Teichmüller mapping. Comput. Methods Appl. Mech. Eng. 311, 41–55.

Oosterlee, C.W., Washio, T., 2000. Krylov subspace acceleration of nonlinear multigrid with application to recirculating flows. SIAM J. Sci. Comput. 21, 1670–1690.

Pan, M., Chen, F., 2019. Low-rank parameterization of volumetric domains for isogeometric analysis. Comput. Aided Des. 114, 82–90.

Pan, M., Chen, F., 2022. Constructing planar domain parameterization with hb-splines via quasi-conformal mapping. Comput. Aided Geom. Des. 97, 102133.

Pan, M., Chen, F., Tong, W., 2018. Low-rank parameterization of planar domains for isogeometric analysis. Comput. Aided Geom. Des. 63, 1–16.

Pan, M., Chen, F., Tong, W., 2020. Volumetric spline parameterization for isogeometric analysis. Comput. Methods Appl. Mech. Eng. 359, 112769.

Pan, Q., Rabczuk, T., Yang, X., 2021. Subdivision-based isogeometric analysis for second order partial differential equations on surfaces. Comput. Mech. 68, 1205–1221.

Peng, Y., Deng, B., Zhang, J., Geng, F., Qin, W., Liu, L., 2018. Anderson acceleration for geometry optimization and physics simulation. ACM Trans. Graph. 37, 1–14.

Pilgerstorfer, E., Jüttler, B., 2014. Bounding the influence of domain parameterization and knot spacing on numerical stability in isogeometric analysis. Comput. Methods Appl. Mech. Eng. 268, 589–613.

Pollock, S., Rebholz, L.G., 2021. Anderson acceleration for contractive and noncontractive operators. IMA J. Numer. Anal. 41, 2841–2872.

Pollock, S., Rebholz, L.G., Xiao, M., 2019. Anderson-accelerated convergence of Picard iterations for incompressible Navier–Stokes equations. SIAM J. Numer. Anal. 57, 615–637.

Pulay, P., 1980. Convergence acceleration of iterative sequences. The case of SCF iteration. Chem. Phys. Lett. 73, 393–398.

Shepherd, K.M., Gu, X.D., Hiemstra, R.R., Hughes, T.J., 2022a. Quadrilateral layout generation and optimization using equivalence classes of integral curves: theory and application to surfaces with boundaries. J. Mech. 38, 128–155.

Shepherd, K.M., Gu, X.D., Hughes, T.J., 2022b. Isogeometric model reconstruction of open shells via Ricci flow and quadrilateral layout-inducing energies. Eng. Struct. 252, 113602.

Sterck, H.D., 2012. A nonlinear GMRES optimization algorithm for canonical tensor decomposition. SIAM J. Sci. Comput. 34, A1351–A1379.

Sterck, H.D., He, Y., 2021. On the asymptotic linear convergence speed of Anderson acceleration, Nesterov acceleration, and nonlinear GMRES. SIAM J. Sci. Comput. 43, S21–S46.

Su, J.P., Fu, X.M., Liu, L., 2019. Practical foldover-free volumetric mapping construction. In: Computer Graphics Forum. Wiley Online Library, pp. 287–297.

Toth, A., Ellis, J.A., Evans, T., Hamilton, S., Kelley, C., Pawlowski, R., Slattery, S., 2017. Local improvement results for Anderson acceleration with inaccurate function evaluations. SIAM J. Sci. Comput. 39, S47–S65.

Toth, A., Kelley, C.T., 2015. Convergence analysis for Anderson acceleration. SIAM J. Numer. Anal. 53, 805–819.

Ugalde, I.A., Mederos, V.H., Sánchez, P.B., Flores, G.G., 2018. Injectivity of B-spline biquadratic maps. Comput. Methods Appl. Mech. Eng. 341, 586–608.

Van der Vorst, H.A., Vuik, C., 1994. GMRESR: a family of nested GMRES methods. Numer. Linear Algebra Appl. 1, 369–386.

Vuik, C., 1993. Solution of the discretized incompressible Navier-Stokes equations with the GMRES method. Int. J. Numer. Methods Fluids 16, 507–523.

Walker, H.F., Ni, P., 2011. Anderson acceleration for fixed-point iterations. SIAM J. Numer. Anal. 49, 1715–1735.

Wang, D., He, Y., De Sterck, H., 2021. On the asymptotic linear convergence speed of Anderson acceleration applied to admm. J. Sci. Comput. 88, 1–35.

Wang, S., Ren, J., Fang, X., Lin, H., Xu, G., Bao, H., Huang, J., 2022a. IGA-suitable planar parameterization with patch structure simplification of closed-form polysquare. Comput. Methods Appl. Mech. Eng. 392, 114678.

Wang, X., Ma, W., 2021. Smooth analysis-suitable parameterization based on a weighted and modified Liao functional. Comput. Aided Des. 140, 103079.

Wang, X., Qian, X., 2014. An optimization approach for constructing trivariate B-spline solids. Comput. Aided Des. 46, 179–191.

Wang, Z., Cao, J., Wei, X., Zhang, Y.J., 2022b. TCB-spline-based isogeometric analysis method with high-quality parameterizations. Comput. Methods Appl. Mech. Eng. 393, 114771.

Xiao, S., Kang, H., Fu, X.M., Chen, F., 2018. Computing IGA-suitable planar parameterizations by PolySquare-enhanced domain partition. Comput. Aided Geom. Des. 62, 29–43.

Xie, J., Xu, J., Dong, Z., Xu, G., Deng, C., Mourrain, B., Zhang, Y.J., 2020. Interpolatory Catmull-Clark volumetric subdivision over unstructured hexahedral meshes for modeling and simulation applications. Comput. Aided Geom. Des. 80, 101867.

Xu, G., Kwok, T.H., Wang, C.C., 2017. Isogeometric computation reuse method for complex objects with topology-consistent volumetric parameterization. Comput. Aided Des. 91, 1–13.

Xu, G., Li, B., Shu, L., Chen, L., Xu, J., Khajah, T., 2019. Efficient r-adaptive isogeometric analysis with Winslow's mapping and monitor function approach. J. Comput. Appl. Math. 351, 186–197.

Xu, G., Li, M., Mourrain, B., Rabczuk, T., Xu, J., Bordas, S.P., 2018. Constructing IGA-suitable planar parameterization from complex CAD boundary by domain partition and global/local optimization. Comput. Methods Appl. Mech. Eng. 328, 175–200.

Xu, G., Mourrain, B., Duvigneau, R., Galligo, A., 2011. Parameterization of computational domain in isogeometric analysis: methods and comparison. Comput. Methods Appl. Mech. Eng. 200, 2021–2031.

Xu, G., Mourrain, B., Duvigneau, R., Galligo, A., 2013a. Analysis-suitable volume parameterization of multi-block computational domain in isogeometric applications. Comput. Aided Des. 45, 395–404.

Xu, G., Mourrain, B., Duvigneau, R., Galligo, A., 2013b. Constructing analysis-suitable parameterization of computational domain from CAD boundary by variational harmonic method. J. Comput. Phys. 252, 275–289.

Xu, G., Mourrain, B., Duvigneau, R., Galligo, A., 2013c. Optimal analysis-aware parameterization of computational domain in 3D isogeometric analysis. Comput. Aided Des. 45, 812–821.

Xu, G., Mourrain, B., Wu, X., Chen, L., Hui, K.C., 2015. Efficient construction of multi-block volumetric spline parameterization by discrete mask method. J. Comput. Appl. Math. 290, 589–597.

Zhang, C., Chai, S., Liu, L., Fu, X.M., 2021. Quad meshing with coarse layouts for planar domains. Comput. Aided Des. 140, 103084.

Zhang, J., O'Donoghue, B., Boyd, S., 2020. Globally convergent type-I Anderson acceleration for nonsmooth fixed-point iterations. SIAM J. Optim. 30, 3170–3197.

Zhang, J., Peng, Y., Ouyang, W., Deng, B., 2019. Accelerating ADMM for efficient simulation and optimization. ACM Trans. Graph. 38, 1–21.

Zhang, Y., Wang, W., Hughes, T.J., 2012. Solid T-spline construction from boundary representations for genus-zero geometry. Comput. Methods Appl. Mech. Eng. 249–252, 185–197.

Zhang, Y., Wang, W., Hughes, T.J., 2013. Conformal solid T-spline construction from boundary T-spline representations. Comput. Mech. 51, 1051–1059.

Zheng, Y., Chen, F., 2022. Volumetric parameterization with truncated hierarchical B-splines for isogeometric analysis. Comput. Methods Appl. Mech. Eng. 401, 115662.