High-quality planar NURBS parameterization of computational domain in IGA via control points and weights optimization

纪野, 于滢滢, 王梦云, 朱春钢

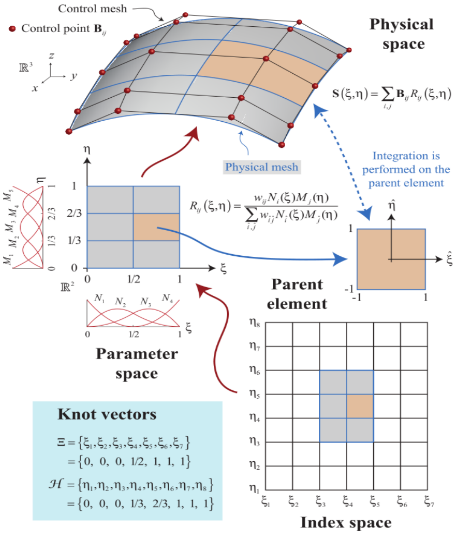
Institute of Computational Science, School of Mathematical Sciences

Key Laboratory for Computational Mathematics and Data Intelligence of Liaoning Province

Dalian University of Technology, Dalian 116024

May 30, 2021

Catalogue



- Proposed by T.J.R. Hughes et al., 2005.
- KEY IDEA: approximate the physical fields with the same basis functions as that used to generate the CAD model.
- Advantages:
  - Integration of design and analysis
  - Exact and efficient geometry
  - Simplified mesh refinement
  - High order continuous field
  - Superior approximation properties
- Very broad applications: such as shell analysis, fluid–structure interaction, and shape optimization.

NURBS Parameterization for IGA

- Similar to mesh generation in FEA, constructing analysis-suitable parameterization is a crucial step in IGA.
- **Motivation**: only the internal control points are considered as variables by previous research. We consider **truly rational parameterization** in this work.

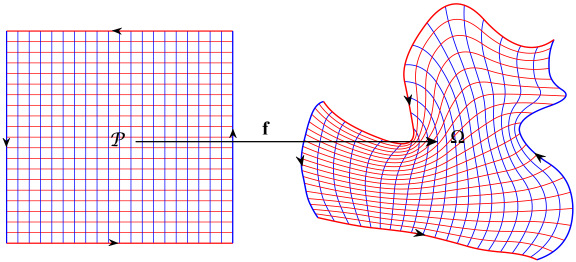


Fig. 1 Parameterization problem.

Catalogue

Related work

- Effect of parameterization quality on subsequent analysis: E. Cohen et al. 2010, G. Xu et al. 2013a, E. Pilgerstorfer et al. 2014.
- Planar domain parameterization:
  - **single-patch: Xu et al. 2011a 2011b 2013b 2019, Gravesen et al. 2014, Choi et al. 2015, Nian and Chen 2016, Hinz et al. 2018, Ugalde et al. 2018, Pan et al. 2018, Zheng et al. 2019.**
  - multi-patch: Xu et al. 2015, Buchegger et al. 2018, Xu et al. 2018, Xiao et al. 2018, Kapl et al. 2017a 2017b 2018 2019, Blidia et al. 2020.
- Volumetric parameterization:
  - single-block: Martin et al. 2009, Aigner et al.2009, Nguyen et al. 2014, Wang and Qian 2014, Xu et al. 2014, Pan et al. 2020, Liu et al. 2020, Yuan et al. 2021.
  - multi-block: Xu et al. 2013 2017, Hu and Lin et al. 2017 2019, Chen et al. 2019, Haberleitner 2019.
- Non-standard B-splines or NURBS: such as C^1 Powell-Sabin splines, toric patches, THB-splines, T-splines, PHT-splines, and Catmull-Clark volumetric subdivision.

Catalogue

A sufficient condition for injective mapping

Lemma 1 (Kestelman, 1971)

Suppose that $\mathbf{f}$ *is a* C^1 *parameterization from parametric domain* $\mathcal{P} = [0,1]^2$ *to computational domain* Ω*, satisfying* $\mathbf{f}|_{\partial\mathcal{P}}$ *is injective. Then* $\mathbf{f}$ *is injective if its Jacobian* $\mathbf{J}$ *is invertible at all points of* $\mathcal{P}$*.*

From Lemma 1, if the Jacobian determinant $\det \mathbf{J}$ have the same sign on the whole parametric domain $\mathcal{P}$, then the differentiable parameterization $\mathbf{f}$ is injective.

Planar NURBS parameterization

- NURBS parameterization:

$$\begin{aligned} \mathbf{f}: \quad \mathcal{P} \quad &\rightarrow \quad \Omega \\ (u,v)^{\mathrm{T}} \mapsto (x,y)^{\mathrm{T}} &= \frac{\sum_{i=0}^{n}\sum_{j=0}^{m}\omega_{ij}\mathbf{P}_{ij}N_i^p(u)N_j^q(v)}{\sum_{i=0}^{n}\sum_{j=0}^{m}\omega_{ij}N_i^p(u)N_j^q(v)}. \end{aligned} \tag{1}$$

- Partial derivatives:

$$\begin{aligned} \frac{\partial \mathbf{f}}{\partial u} &= \frac{p}{W^2}(\mathbf{P}_u^{\omega}W - \mathbf{P}^{\omega}W_u), \\ \frac{\partial \mathbf{f}}{\partial v} &= \frac{q}{W^2}(\mathbf{P}_v^{\omega}W - \mathbf{P}^{\omega}W_v). \end{aligned} \tag{2}$$

Jacobian determinant

- Utilizing the product properties of B-splines [Mϕrken, 1991], the Jacobian determinant $\det \mathbf{J}$ of the NURBS parameterization (1) can be represented in the form as

$$\begin{aligned} \det \mathbf{J}(u,v) &= \left| \frac{\partial \mathbf{f}}{\partial u}, \frac{\partial \mathbf{f}}{\partial v} \right| \\ &= \frac{pq}{W^4} \left| \mathbf{P}_u^{\omega} W - \mathbf{P}^{\omega} W_u, \mathbf{P}_v^{\omega} W - \mathbf{P}^{\omega} W_v \right| \\ &= \frac{pq}{W^4} \sum_{i=0}^{n^J} \sum_{j=0}^{m^J} N_i^{4p-1}(u) N_j^{4q-1}(v) J_{i,j}^N. \end{aligned} \tag{3}$$

A sufficient condition for injective NURBS parameterization

- Since $pq/W^4 > 0$ for $\forall(u,v) \in \mathcal{P}$, the sign of $\det \mathbf{J}$ is entirely determined by the sign of the following term

$$J_D(u,v) = \sum_{i=0}^{n^J} \sum_{j=0}^{m^J} N_i^{4p-1}(u) N_j^{4q-1}(v) J_{i,j}^N. \tag{4}$$

- According to the nonnegativity of B-spline basis functions, the following theorem can be deduced easily from Lemma 1.

Theorem 2

If all the control coefficients $J_{i,j}^N$ in (4) are positive, then the planar NURBS parameterization $\mathbf{f}$ in (1) is injective.

Weaker sufficient conditions

- By taking the knot insertion and/or degree elevation algorithm, we can obtain a series of weaker sufficient injectivity conditions for NURBS parameterizations.
- A better way is to convert the B-spline representation (4) into Bézier form with the Bézier extraction technique:

$$J_D^B(u,v) = \sum_{s_1=0}^{4p-1} \sum_{s_2=0}^{4q-1} B_{s_1}^{4p-1}(u) B_{s_2}^{4q-1}(v) J_{s_1,s_2}^B, \tag{5}$$

Weaker sufficient conditions - continued

where

$$J^B_{s_1,s_2} = \sum_{\substack{i_1+i_2+i_3+i_4=s_1\\0\leqslant i_1\leqslant p-1\\0\leqslant i_2\leqslant p\\0\leqslant i_3\leqslant p\\0\leqslant i_4\leqslant p}} \sum_{\substack{j_1+j_2+j_3+j_4=s_2\\0\leqslant j_1\leqslant q\\0\leqslant j_2\leqslant q\\0\leqslant j_3\leqslant q-1\\0\leqslant j_4\leqslant q}} \gamma_{s_1,s_2}\left|\mathbf{\Gamma}^u_{i_1,i_2,j_1,j_2},\mathbf{\Gamma}^v_{i_3,i_4,j_3,j_4}\right|, \tag{6}$$

$$\gamma_{s_1,s_2} = \frac{\binom{p-1}{i_1}\binom{p}{i_2}\binom{p}{i_3}\binom{p}{i_4}\binom{q-1}{j_1}\binom{q}{j_2}\binom{q}{j_3}\binom{q}{j_4}}{\binom{4p-1}{s_1}\binom{4q-1}{s_2}}.$$

$$\mathbf{\Gamma}^u_{i_1,i_2,j_1,j_2} = \omega_{i_2,j_2}(\omega_{i_1+1,j_1}(\mathbf{P}_{i_1+1,j_1}-\mathbf{P}_{i_2,j_2})-\omega_{i_1,j_1}(\mathbf{P}_{i_1,j_1}-\mathbf{P}_{i_2,j_2})),$$

and

$$\mathbf{\Gamma}^v_{i_3,i_4,j_3,j_4} = \omega_{i_4,j_4}(\omega_{i_3,j_3+1}(\mathbf{P}_{i_3,j_3+1}-\mathbf{P}_{i_4,j_4})-\omega_{i_3,j_3}(\mathbf{P}_{i_3,j_3}-\mathbf{P}_{i_4,j_4})).$$

Necessary conditions for injective NURBS parameterization

- Bézier function interpolates its corner control coefficients.
- Necessary condition of the injective parameterization can be obtained.

Theorem 3

If the parameterization $\mathbf{f}$ *in (1) is injective, then the corner Bézier control coefficients* $J^B_{0,0}, J^B_{0,4q-1}, J^B_{4p-1,0}, J^B_{4p-1,4q-1}$ *in formula (6) are non-negative.*

Algorithm for injectivity checking

Algorithm 1 Checking the injectivity of NURBS parameterization

Input: Control points $\mathbf{P}_{ij}$, weights ω_{ij}, degrees p, q, and knot vectors $\mathbf{U}, \mathbf{V}$ of the NURBS parameterization $\mathbf{f}$, maximum iterations K_{max}.

Output: Injectivity of $\mathbf{f}$.

1: Compute the extracted Bézier patches by the Bézier extraction technique, and set $k = 0$;
2: For each new Bézier patch, compute the Bézier control coefficients $J^B_{s_1,s_2}$ in (6);
3: If all the corner Bézier control coefficients $J^B_{0,0}, J^B_{0,4q-1}, J^B_{4p-1,0}, J^B_{4p-1,4q-1}$ are non-negative, go to Step 4; otherwise, return 'Invalid';
4: If all Bézier control coefficients $J^B_{s_1,s_2}$ are positive, return 'Valid';
5: If $k > K_{max}$, return 'Maximum iterations has been reached';
6: Refine each Bézier patch with non-positive coefficients into four subpatches by de Casteljau algorithm;
7: Set $k = k + 1$, and go to Step 2.

Catalogue

General constraint optimization framework

- General constraint optimization framework:

$$\begin{aligned} &\arg\min_{\mathbf{x}} E(\mathbf{f}(\mathbf{x})) \\ &s.t. \quad J_{s_1,s_2} \geq \delta, \end{aligned} \tag{7}$$

where $\mathbf{x}$ are optimization variables, $E(\mathbf{f}(\mathbf{x}))$ is an energy functional to characterize the geometric property of the parameterization, such as orthogonality, uniformity, and low distortion, and δ is a positive threshold.

- Since the constraints must be recomputed during each iteration, direct solving problem (7) is computationally unacceptable since forming the control coefficients J_{s_1,s_2} is very costly both in time and memory.
- We propose an three-step unconstrained optimization approach to deal with this problem.

Step 1. Initialization

- Simply set all the internal weights to 1 in the initialization stage.
- Next, the internal control points $\mathbf{P}_{ij}(i = 1, 2, \cdots, n-1, j = 1, 2, \cdots, m-1)$ are obtained by solving the following unconstrained quadratic programming problem

$$\arg\min_{\mathbf{P}_{ij}} \iint_{\mathcal{P}} ||\Delta \mathbf{f}||^2 d\mathcal{P}, \tag{8}$$

where $\Delta = \frac{\partial^2}{\partial u^2} + \frac{\partial^2}{\partial v^2}$ is the Laplace operator.

- No guarantee on free of self-intersection.

Step 2. Eliminating foldovers

- **Eliminating foldovers (E)** by solving the following unconstrained optimization problem:

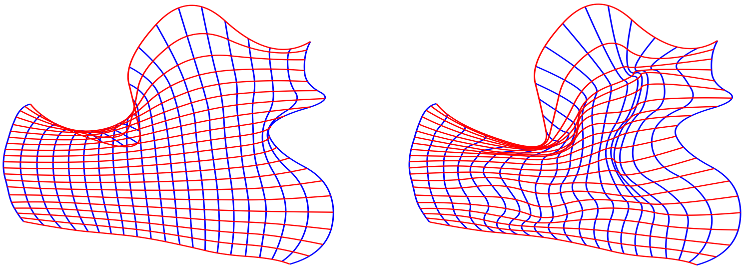
$$\arg\min_{\mathbf{P}_{ij}} \iint_{\mathcal{P}} ReLU(\delta - \det \mathbf{J})d\mathcal{P}. \tag{9}$$

where δ is a user-specified threshold, and $ReLU$ stands for the Rectified Linear Unit, i.e.,

$$ReLU(x) = \begin{cases} 0, & x \leq 0, \\ x, & x > 0. \end{cases}$$

Step 2. Eliminating foldovers - continued

- The quality of the parameterization is still poor.
- The next task is to improve the parameterization quality.



(a) Initialization (b) Eliminating foldovers

Fig. 2 Initialization and eliminating foldovers.

Step 3. Improving parameterization quality

- **Winslow's functional (W)**:

$$E^W(\mathbf{f}; \mathbf{P}_{ij}, \omega_{ij}) = \iint_{\mathcal{P}} \frac{tr(\mathbf{g})}{\det \mathbf{J}} d\mathcal{P} \tag{10}$$

where $tr(\mathbf{g})$ is the trace of the first fundamental form $\mathbf{g}$.

Remarks

- The minimum of Winslow's functional provides a diffeomorphic mapping between the parametric domain and the computational domain which is ensured by the Radó-Kneser-Choquet theorem.
- The integral term approaches infinity as $\det \mathbf{J}$ approaches to zero, which can suppress foldovers effectively.

Corrected Winslow's functional

- Since the change of the objective function is discontinuous during the numerical optimization process, the value of the objective function may step over zero directly.
- **Corrected Winslow's functional (CW)**:

$$E^{CW}(\mathbf{f}; \mathbf{P}_{ij}, \omega_{ij}) = \iint_{\mathcal{P}} \frac{tr(\mathbf{g})}{ReLU(\det \mathbf{J}) + \varepsilon} d\mathcal{P}, \tag{11}$$

  where ε is a small positive threshold to prevent dividing by zero.
- When the Jacobian determinant $\det \mathbf{J}$ is non-positive, the denominator of the integral term is equal to ε. This cause quite tremendous value of $E^{CW}(\mathbf{f}; \mathbf{P}_{ij}, \omega_{ij})$, which penalizes invalid parameterization.

Uniformity functional

- For the most uniform case, Jacobian determinan $\det \mathbf{J}$ is equal to S (S is the area of the given computational domain) anywhere.
- Given the boundary representation in four NURBS curves form, the area of the computational domain can be obtained easily by Green's formula.
- **Uniformity functional (U)**:

$$E^{uniform}(\mathbf{f}; \mathbf{P}_{ij}, \omega_{ij}) = \iint_{\mathcal{P}} \left(\frac{\det \mathbf{J}}{S} - 1 \right)^2 d\mathcal{P}, \qquad (12)$$

Optimization model

- Objective functional:

$$E(\mathbf{f};\mathbf{P}_{ij},\omega_{ij}) = E^{CW}(\mathbf{f};\mathbf{P}_{ij},\omega_{ij}) + \lambda E^{uniform}(\mathbf{f};\mathbf{P}_{ij},\omega_{ij}), \tag{13}$$

where λ is a positive weight for balance between the corrected Winslow's functional and the uniformity functional.

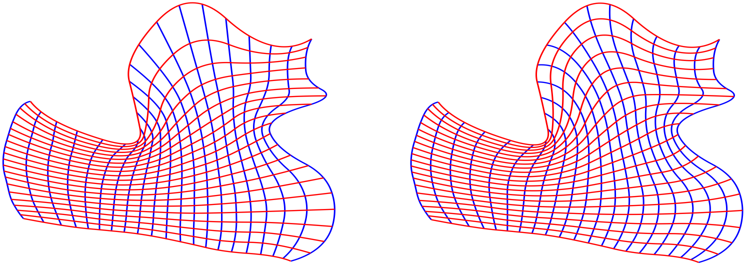
- Solving the two following sub-problems alternately:
  (i) Fix the internal weights, and update the internal control points, i.e.,

$$\arg\min_{\mathbf{P}_{ij}} E(\mathbf{f};\mathbf{P}_{ij},\omega_{ij}); \tag{14}$$

  (ii) Fix the internal control points, and update the corresponding weights, i.e.,

$$\arg\min_{\omega_{ij}} E(\mathbf{f};\mathbf{P}_{ij},\omega_{ij}). \tag{15}$$

With/without uniformity functional



(a) Without uniformity functional (b) With uniformity functional

Fig. 3 Comparison of parameterization with/without uniformity functional.

Algorithm for constructing planar NURBS parameterization

Algorithm 2 Constructing high-quality planar NURBS parameterization by adjusting control points and weights

Input: Four boundary representations of the domain Ω in NURBS form, and the parameters $\delta, \varepsilon, \lambda, \varepsilon_0, N_{max}$.
Output: Internal control points, weights, and the corresponding planar NURBS parameterization.

1: Set all the internal weights to 1, and construct the initial internal control points by solving the linear system obtained by the harmonic map (8);
2: Eliminate foldovers by solving the optimization problem (9);
3: Set $k = 0$;
4: **repeat:**
5: Fix weights ω_{ij}, solving the optimization problem (14) to obtain the internal control points $\mathbf{P}_{ij}$, and evaluate the value E_1 of $E(\mathbf{f}; \mathbf{P}_{ij}, \omega_{ij})$;
6: Fix control points $\mathbf{P}_{ij}$, solving the optimization problem (15) to obtain the internal weights ω_{ij}, and evaluate the value E_2 of $E(\mathbf{f}; \mathbf{P}_{ij}, \omega_{ij})$;
7: Set $k = k + 1$;
8: **until** $k > N_{max}$ or $|E_2 - E_1| < \varepsilon_0$;

Catalogue

Numerical examples and comparisons

Implementation details

- MATLAB (MATLAB R2020a) + C++ (Visual Studio 2017)
- Optimizer: L-BFGS in NLopt [1].
- Gaussian quadrature rules for integral items and the MATLAB backslash divide command for solving linear systems.
- All parameters involved in our algorithm are set as default values.
- The maximum number $K_{max} = 5$ in Algorithm 1; the parameters $\delta = 0.05S$ in (9) and $\varepsilon = 10^{-10}$ in (11); the weight $\lambda = 1$ in (13); and the threshold $\varepsilon_0 = 0.01$ and the maximum iterations $N_{max} = 5$ for the stopping criteria in Algorithm 2.

[1] S.G. Johnson, The NLopt nonlinear-optimization package, http://github.com/stevengj/nlopt.

Quality metrics for parameterization quality

- Scaled Jacobian (orthogonality):

$$J_s = \frac{\det \mathbf{J}}{\|\mathbf{f}_u\| \|\mathbf{f}_v\|}. \tag{16}$$

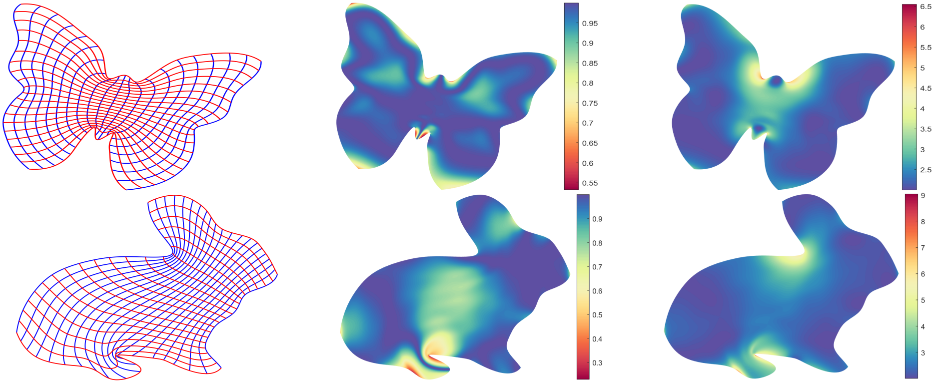
- Condition number (conformal distortion):

$$k(\mathbf{J}) = \|\mathbf{J}\|_F \|\mathbf{J}^{-1}\|_F, \tag{17}$$

where $\|\mathbf{J}\|_F = (tr(\mathbf{J}^\mathrm{T}\mathbf{J}))^{\frac{1}{2}}$ is the Frobenius norm of $\mathbf{J}$.

- A dense sampling (501×501 sample points).

The resulting parameterizations and quality metrics

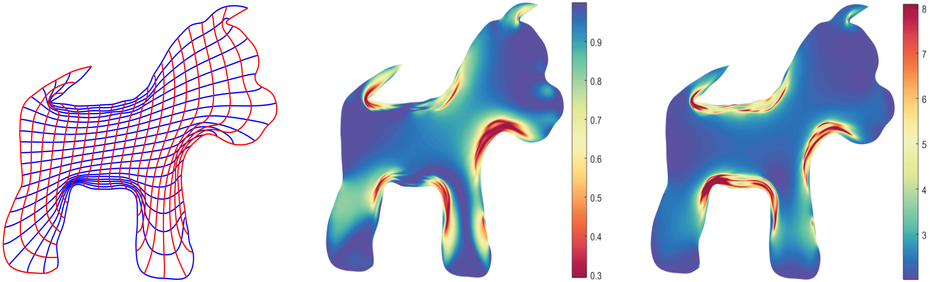


(a) Models (b) Scaled Jacobian (c) Condition number

Fig. 4 Parameterization results and quality metrics of Butterfly and Rabbit model.

Effect of weights

- Whether the use of weights really improves the quality of the parameterizations?
- Test the involved models while keeping all the internal weights equal to one.



(a) Parameterization (b) Scaled Jacobian (c) Condition number

Fig. 5 Parameterization and quality metrics of the Dog model with fixed internal weights.

Table 1 parameterization quality w.r.t. various methods

model	p, q, n, m	method	Scaled Jacobian		Condition number	
			average	min	average	max
Rabbit	3,3,13,10	Fixed weights	0.9152	-0.0090	2.5671	374.6158
		Ours	**0.9159**	**0.2968**	**2.5584**	**6.7861**
Dog	3,3,30,30	Fixed weights	0.8331	0.1720	3.6472	17.7726
		Ours	**0.8420**	**0.2335**	**3.2194**	**9.2387**

- Constructs a poor-quality parameterization in the Dog model.
- Fails to obtain a valid parameterization in the Rabbit model.
- Optimizing the weights indeed has a positive effect on the injectivity and quality of parameterization.

Comparisons

We bring the proposed method into comparison with several state-of-the-art parameterization methods:

1. Nonlinear Constrained Optimization method (NCO) [1],
2. Variational Harmonic method (VH) [2],
3. Teichmüller mapping method (T-Map) [3], and
4. Low-Rank Quasi-Conformal method (LRQC) [4].

---

[1] G. Xu, B. Mourrain, R. Duvigneau, A. Galligo, Parameterization of computational domain in isogeometric analysis: methods and comparison, Comput. Methods Appl. Mech. Engrg. 200 (23–24) (2011) 2021–2031.

[2] G. Xu, B. Mourrain, R. Duvigneau, A. Galligo, Optimal analysis-aware parameterization of computational domain in 3D isogeometric analysis, Comput. Aided Des. 45 (4) (2013) 812–821.

[3] X.S. Nian, F.L. Chen, Planar domain parameterization for isogeometric analysis based on Teichmüller mapping, Comput. Methods Appl. Mech. Engrg. 311 (2016) 41–55.

[4] M.D. Pan, F.L. Chen, W.H. Tong, Low-rank parameterization of planar domains for isogeometric analysis, Comput. Aided Geom. Design 63 (2018) 1–16.

Comparisons: Injectivity

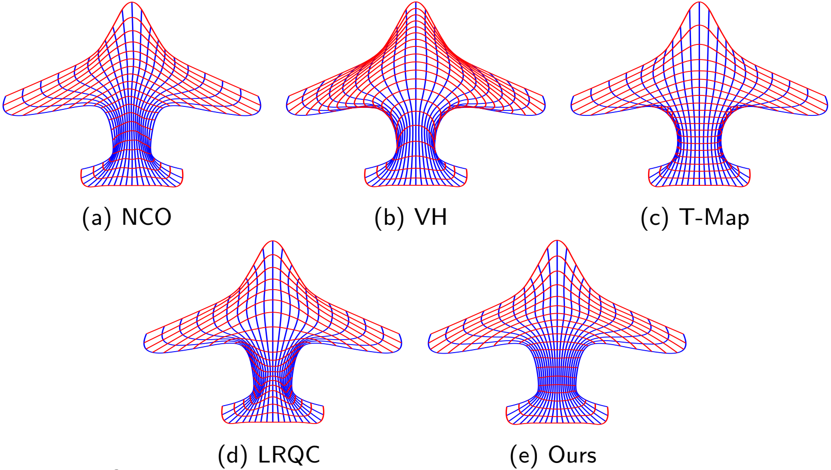
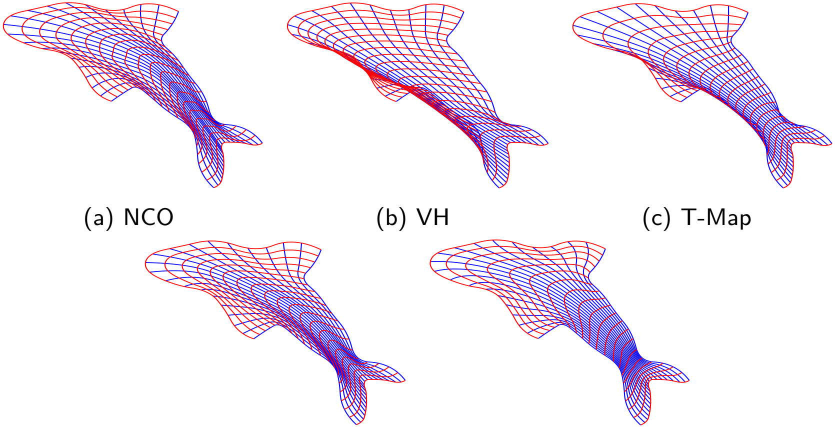


Fig. 6 Parameterization of the Plane model by various methods.

Comparisons: Injectivity - continued

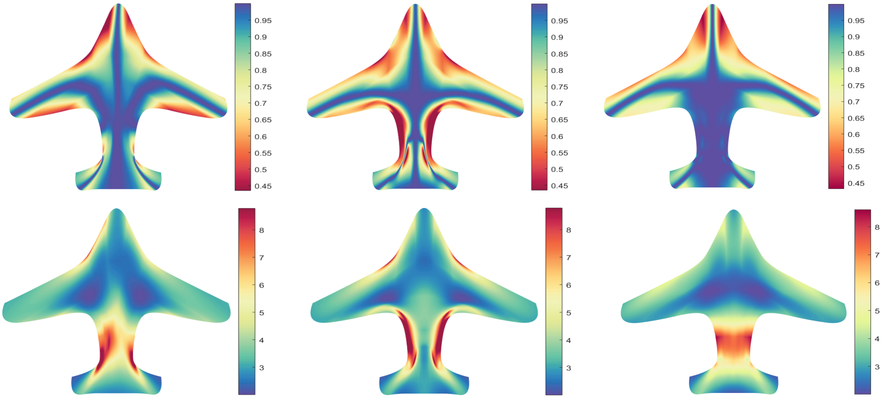


(a) NCO (b) VH (c) T-Map

(d) LRQC (e) Ours

Fig. 7 Parameterization of the Dolphin model by various methods.

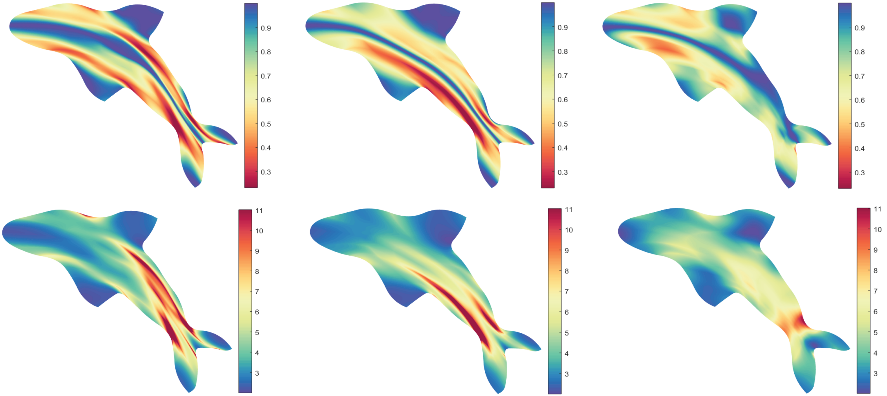
Comparisons: Parameterization quality



(a) NCO (b) LRQC (c) Ours

Fig. 8 Quality metrics of the Plane model by three methods.

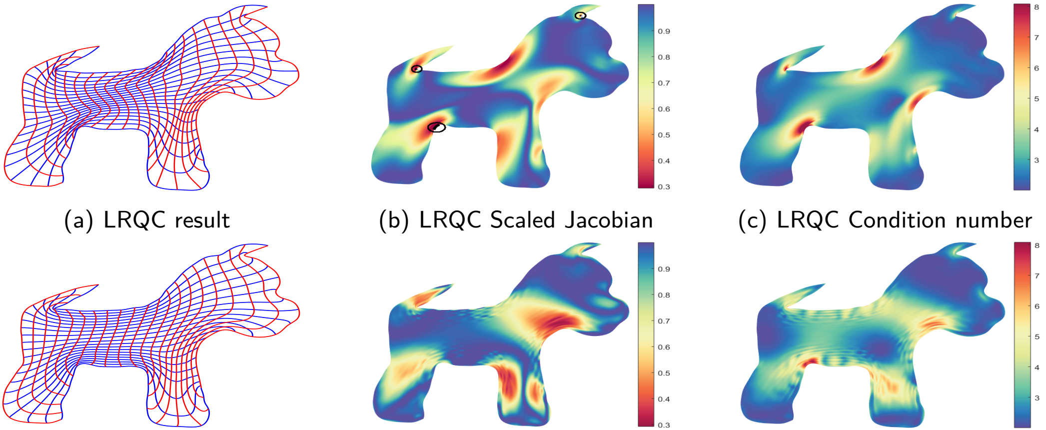
Comparisons: Parameterization quality - continued



(a) NCO (b) LRQC (c) Ours

Fig. 9 Quality metrics of the Dolphin model by three methods.

Comparison between LRQC and our method



(a) LRQC result (b) LRQC Scaled Jacobian (c) LRQC Condition number

(d) Our result (e) Our Scaled Jacobian (f) Our Condition number

Fig. 10 Comparison between LRQC and our method of the Dog model.

Comparison between LRQC and our method - continued

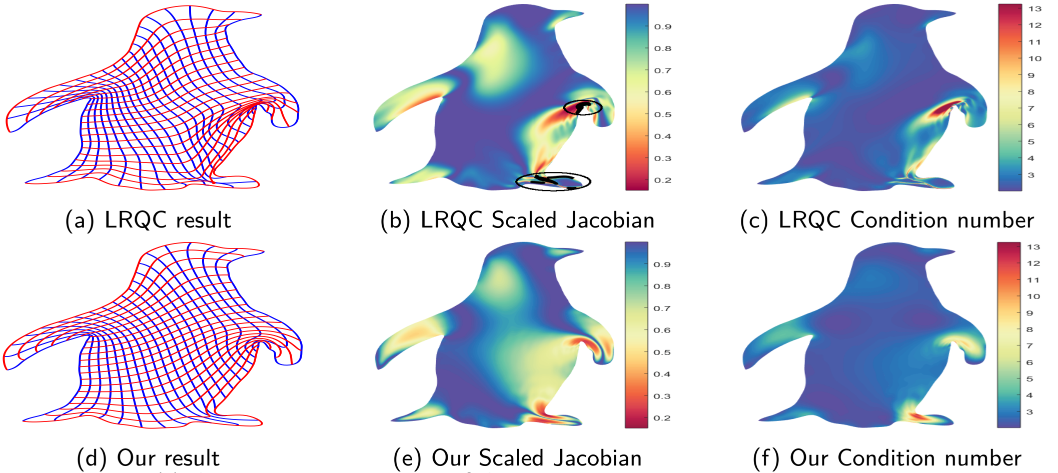


Fig. 11 Comparison between LRQC and our method of the Penguin model.

Table 2 parameterization quality w.r.t. various methods

model	p, q, n, m	method	Scaled Jacobian		Condition number	
			average	min	average	max
Duck	2,2,8,10	NCO	0.8819	0.1160	3.1831	32.2954
		VH	0.7192	-1.0000	13.1134	4.7522e+05
		T-map	0.8709	-1.0000	5.9432	2.5475e+05
		LRQC	0.8850	0.2064	2.9704	16.3875
		Fixed weights	0.9212	0.4122	**2.8992**	**6.8609**
		Ours	**0.9214**	**0.4306**	**2.8992**	6.9994
Butterfly	3,3,12,10	NCO	0.9569	0.3280	2.4956	7.7608
		VH	0.9137	-1.0000	5.8209	6.7621e+04
		T-map	0.9345	-1.0000	7.1372	4.6383e+05
		LRQC	0.9673	0.3206	**2.4184**	14.5630
		Fixed weights	**0.9685**	0.5009	2.4682	**6.1219**
		Ours	**0.9685**	**0.5160**	2.4658	6.7475
Rabbit	3,3,13,10	NCO	0.8957	0.1730	2.8256	11.9616
		VH	0.8923	-1.0000	5.3268	2.0281e+05
		T-map	0.9100	-1.0000	7.5895	6.3149e+05
		LRQC	0.9132	**0.3302**	2.6230	10.1826
		Fixed weights	0.9152	-0.0090	2.5671	374.6158
		Ours	**0.9159**	0.2968	**2.5584**	**6.7861**
Plane	2,2,9,8	NCO	0.8554	0.3360	4.4755	14.8167
		VH	0.6443	-1.0000	18.8121	1.3104e+05
		T-map	0.6879	-1.0000	66.4200	7.1595e+06
		LRQC	0.7249	0.2070	4.7270	15.2799
		Fixed weights	0.9123	**0.4380**	4.1924	**8.5744**
		Ours	**0.9141**	0.4352	**4.1890**	8.7820

Table 3 parameterization quality w.r.t. various methods - continued

model	p,q,n,m	method	Scaled Jacobian		Condition number	
			average	min	average	max
Dolphin	2,2,12,9	NCO	0.5966	0.0823	5.9899	29.2289
		VH	0.4866	-0.6861	78.3844	1.1670e+07
		T-map	0.6561	-0.9523	16.3846	5.5784e+05
		LRQC	0.5961	0.1647	5.8115	17.7545
		Fixed weights	0.7736	**0.2352**	5.0117	11.4117
		Ours	**0.7744**	0.2323	**5.0041**	**11.0161**
Dog	3,3,30,30	NCO	-	-	-	-
		VH	0.4037	-1.0000	21.2539	5.3966e+05
		T-map	-	-	-	-
		LRQC	0.8256	-0.9995	3.4932	2.5641e+03
		Fixed weights	0.8331	0.1720	3.6472	17.7726
		Ours	**0.8488**	**0.2930**	**3.2310**	**8.0818**
Penguin	3,3,30,30	NCO	-	-	-	-
		VH	0.5091	-1.0000	11.9690	2.4950e+05
		T-map	-	-	-	-
		LRQC	0.8252	-1.0000	4.9784	2.8099e+05
		Fixed weights	**0.8491**	0.1430	**2.7048**	14.6129
		Ours	**0.8491**	**0.1515**	2.7050	**13.2554**

Comparisons: Running time

- For practical applications, the efficiency of the algorithm is also a crucial factor.
- Only list the running times of the examples generating valid parameterizations.

Table 4 Running time (in seconds) of various methods

Method	Duck	Butterfly	Rabbit	Plane	Dolphin	Dog	Penguin
NCO	220.27	3567.72	4281.32	304.47	1092.63	-	-
LRQC	1.43	2.99	3.06	1.44	1.77	-	-
Ours	**0.41**	**0.87**	**1.20**	**0.43**	**0.85**	**101.16**	**45.75**

Catalogue

Conclusions and future work

Conclusions

- Several sufficient conditions and a necessary condition for injective NURBS parameterizations.
- An algorithm for the injectivity checking of NURBS parameterizations.
- Parameterization method which alternatingly update weights and control points.
- Numerical examples demonstrate the robustness and efficiency of our method.

Future work

- To high genus computational domains, apply our parameterization approach to multi-patch structures.
- Extend our approach to $3D$ volumetric parameterization.

Thanks for your attention!

Q&A.

jiye@mail.dlut.edu.cn