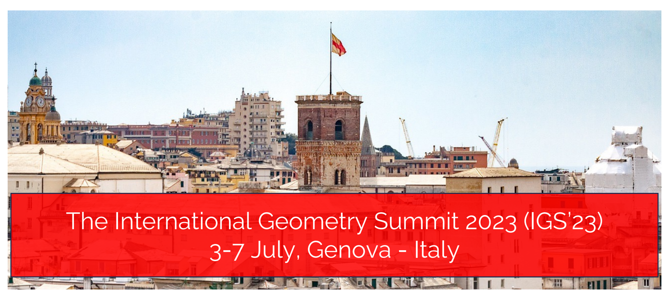
Multi-patch parameterization method for IGA using singular structure of cross-field

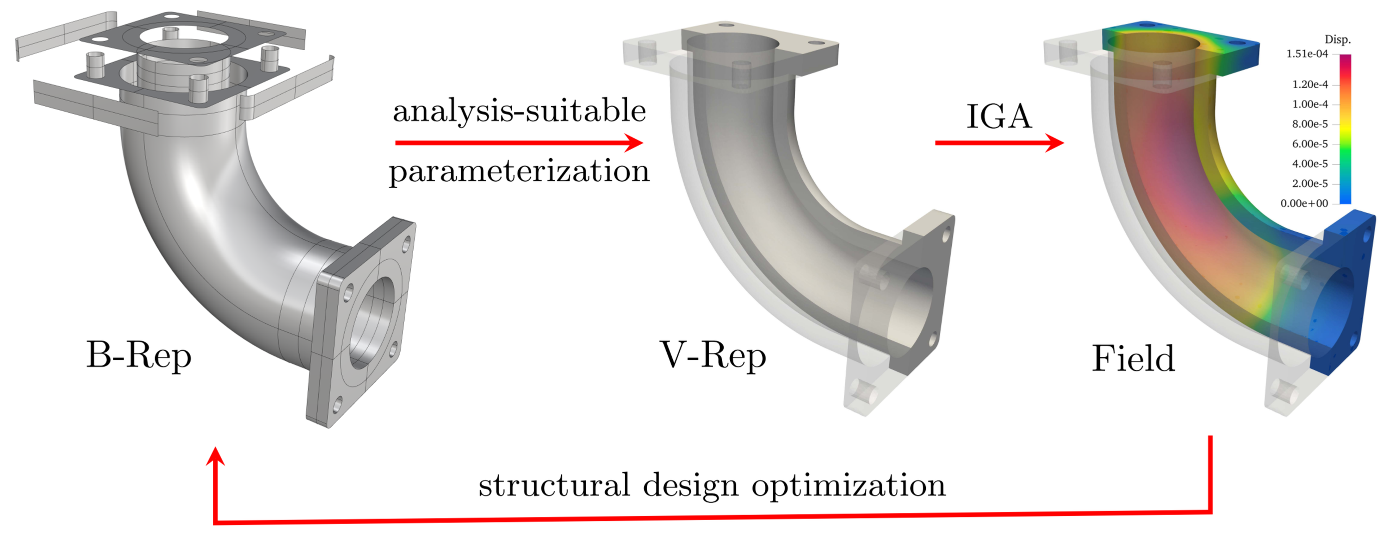
Ye Ji, Yi Zhang, Chun-Gang Zhu

School of Mathematical Sciences, Dalian University of Technology



Problem Definition & Contribution

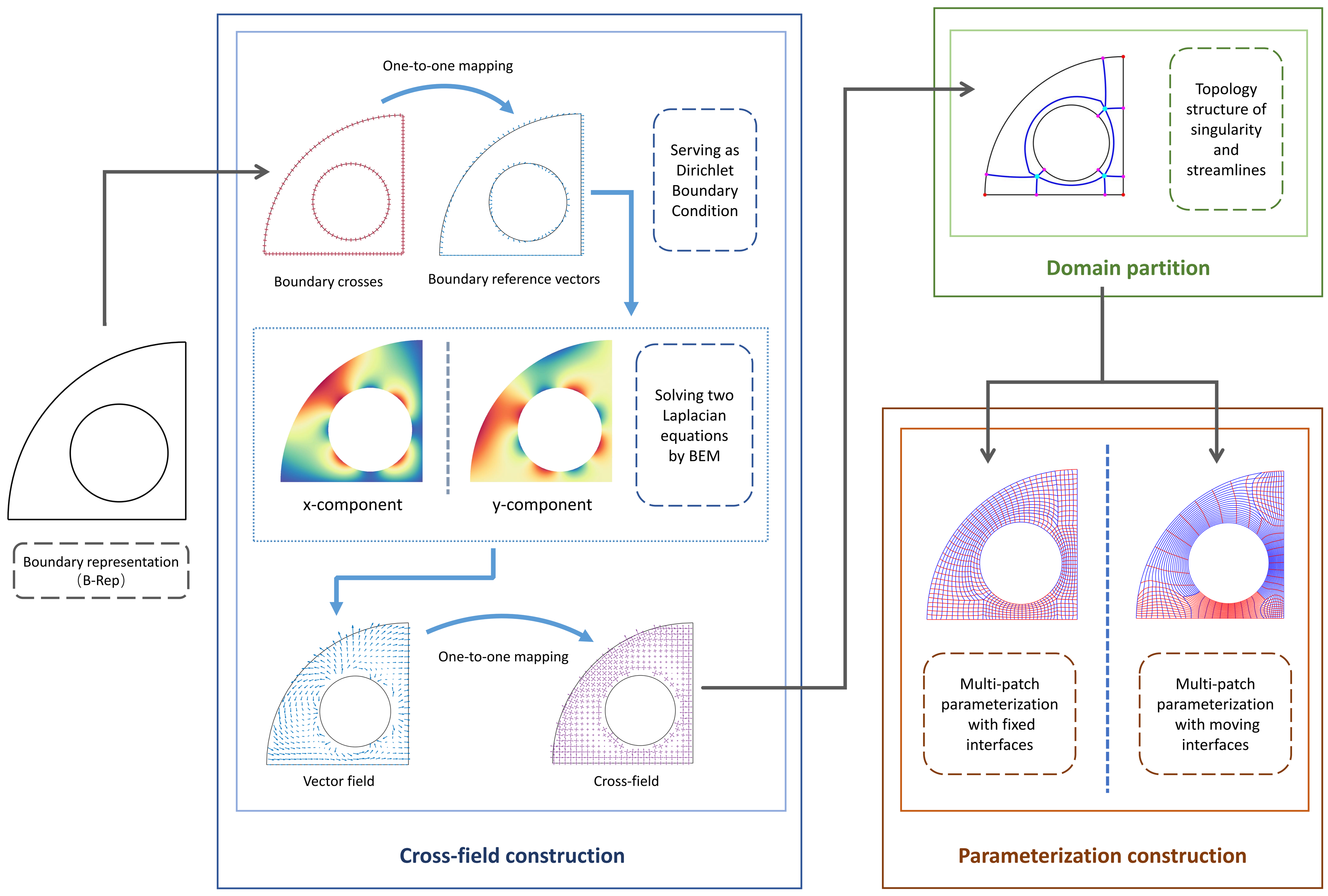
Goal: To construct an **internal spline-based parameterization** $\boldsymbol{x}$ from the boundary representation (B-Rep), such that $\boldsymbol{x}$ **ensures bijectivity** and **exhibits minimal angle and area distortion**.



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

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

Basic Workflow

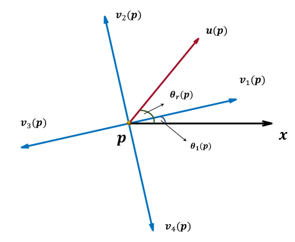


Formulation

Cross-field construction:

- **One-to-one mapping** between a cross $\boldsymbol{v}_k(\boldsymbol{p})$ and its reference vector $\boldsymbol{u}(\boldsymbol{p})$ can be established as:

$$\begin{cases} \|\boldsymbol{u}(\boldsymbol{p})\| = \|\boldsymbol{v}_k(\boldsymbol{p})\|, \\ \theta_r(\boldsymbol{p}) = 4 \cdot \min\{\theta_k(\boldsymbol{p}) : \ k = 1,2,3,4\}. \end{cases}$$



- **Governing equation** propagates the reference vectors $\boldsymbol{u}(\boldsymbol{p})$ from the boundaries to the interior:

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

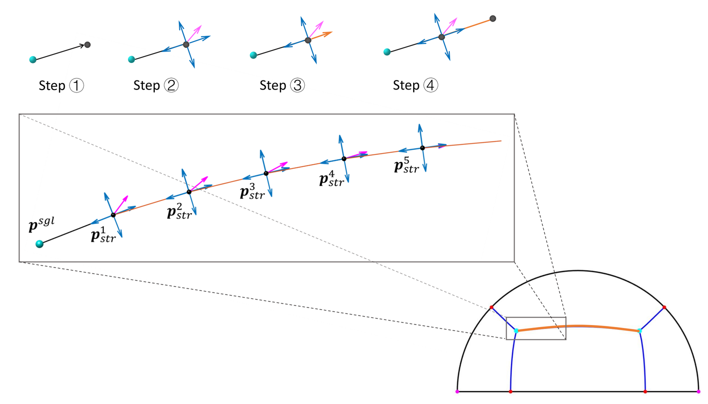
- **Boundary element method (BEM)** is employed since only the B-Rep is known.

Domain partition:

- **Streamline propagation** are calculated by:

$$\boldsymbol{p}_{str}^{s} = \boldsymbol{p}_{str}^{s-1} + \rho dir \boldsymbol{V}^{s-1},$$

where $dir\boldsymbol{V}^s = \frac{\boldsymbol{v}_{i_0}(\boldsymbol{p}_{str}^s)}{\|\boldsymbol{v}_{i_0}(\boldsymbol{p}_{str}^s)\|}$, $i_0 = \underset{k \in \{1,2,3,4\}}{\arg\min} |\theta_0 - \theta_k|$.



Multi-patch parameterization:

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

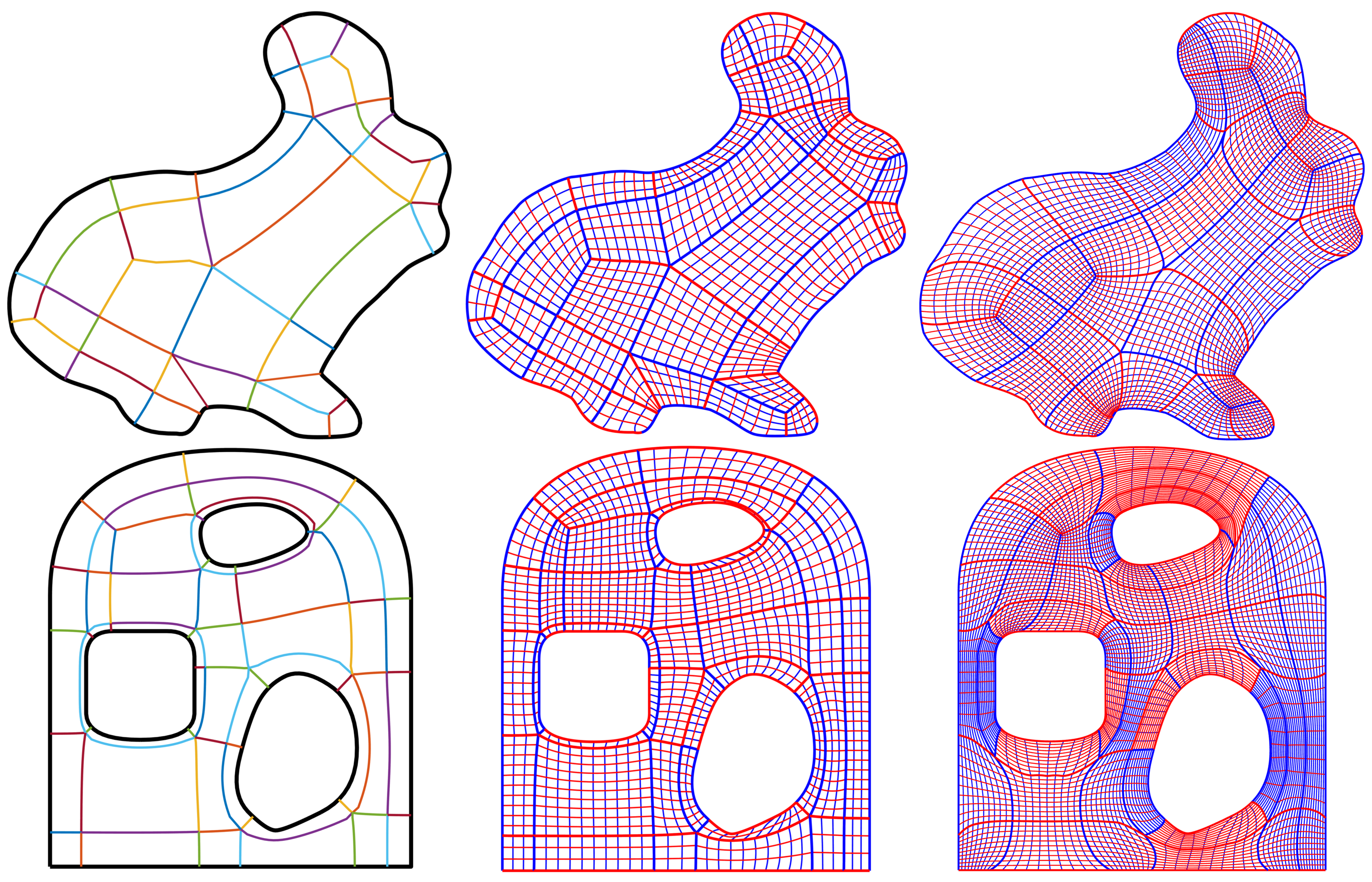
$$\mathscr{E}^{\mathrm{ff}} = \int_{\hat{\Omega}} \max\{0, \delta - |\mathcal{J}|\} \, \mathrm{d}\hat{\Omega}^t,$$

- **Step 3 - Quality improvement:** Objective function

$$\mathscr{E} := \begin{cases} \int_{\hat{\Omega}} tr(\mathcal{J}^{\mathrm{T}}\mathcal{J})/|\mathcal{J}| + \lambda|\mathcal{J}|^2 \mathrm{d}\hat{\Omega}, & \min|\mathcal{J}| > 0, \\ +\infty, & \min|\mathcal{J}| \le 0. \end{cases}$$

Experiments & Results

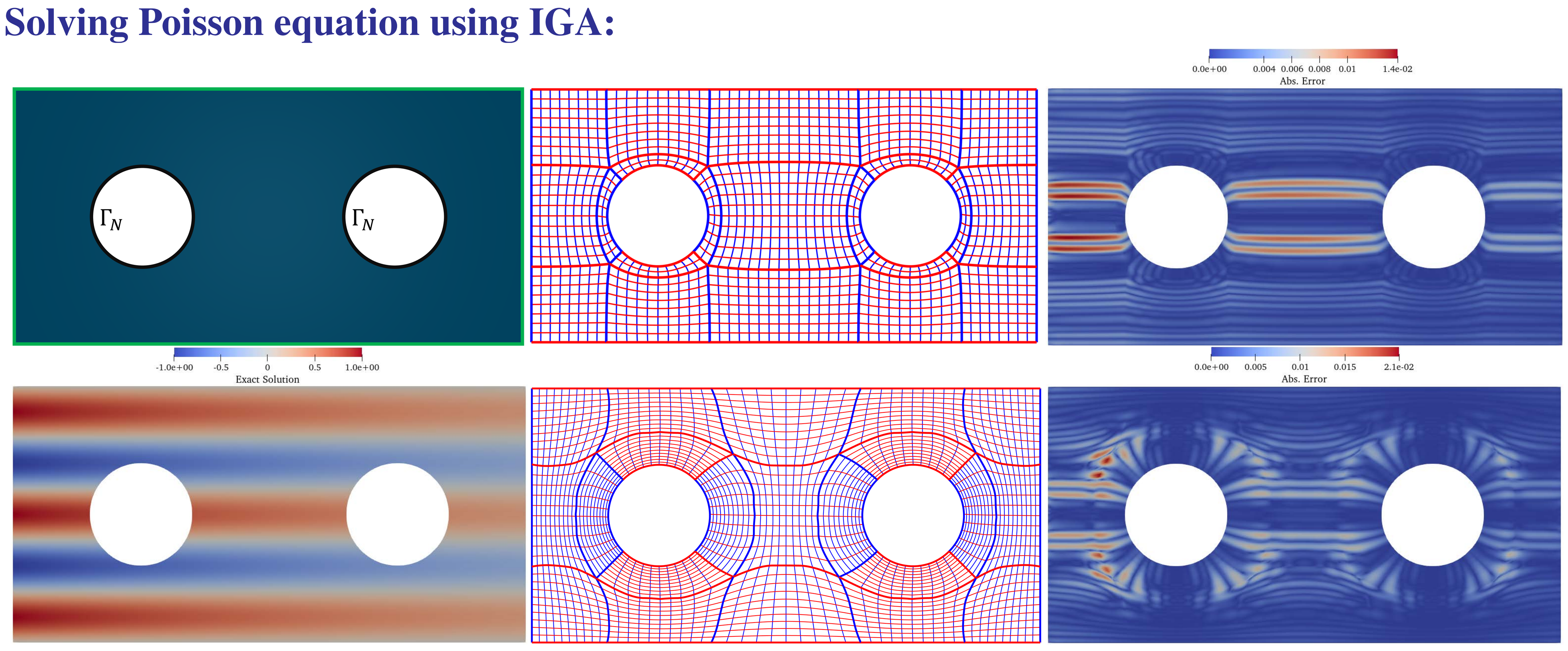
Qualitative results:



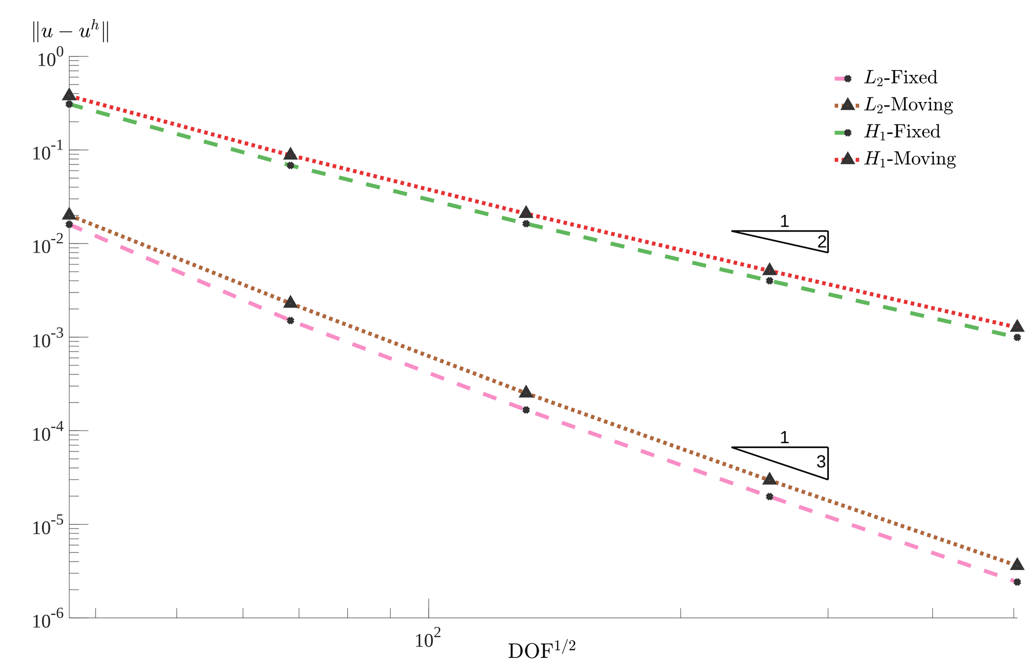
Quantitative results:

Model	#Patch	Method	$\lvert\mathcal{J}\rvert_s$ min.	$\lvert\mathcal{J}\rvert_s$ avg.	$unif.$ min.	$unif.$ avg.	$unif.$ max.	time (sec.)
rabbit	33	discrete Coons	-0.8593	0.9628	0.7030	**0.9410**	1.0524	-
		fixed-interface	**0.2204**	**0.9504**	0.6103	0.9544	**0.9982**	**1.6169**
		moving-interface	0.02918	0.9283	**0.0000**	0.9550	1.0000	23.7138
2 holes	21	smoothness energy	-1	0.8353	0.6981	0.9082	1.0378	-
		fixed-interface	**0.8466**	**0.9892**	0.7354	0.9082	0.9875	**0.2020**
		moving-interface	0.4782	0.9728	**0.7096**	**0.9073**	**0.9674**	0.5404
3 holes	46	discrete Coons	-0.5492	0.9710	0.8008	0.9573	1.0958	-
		fixed-interface	**0.1545**	**0.9716**	0.8007	0.9573	0.9978	**0.4965**
		moving-interface	0.1461	0.9361	**0.6791**	**0.9571**	**0.9968**	5.8342

Solving Poisson equation using IGA:



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



Please feel free to contact me! ;-)

jiyess@outlook.com & https://jiyess.github.io