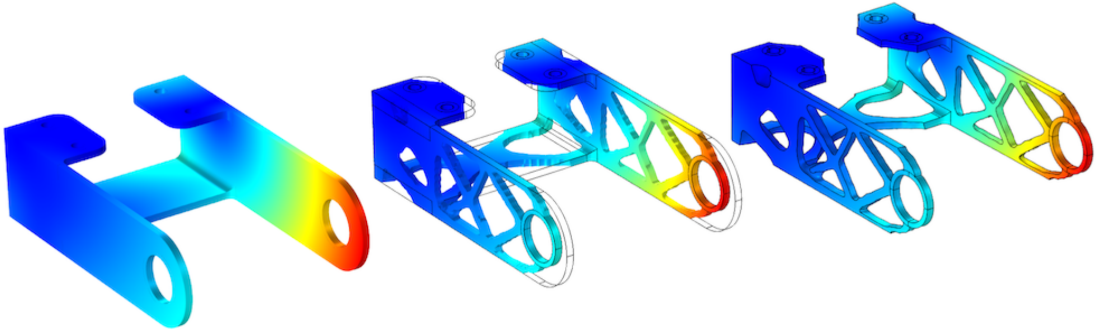
Physics-informed machine learning in design optimization

Matthias Möller

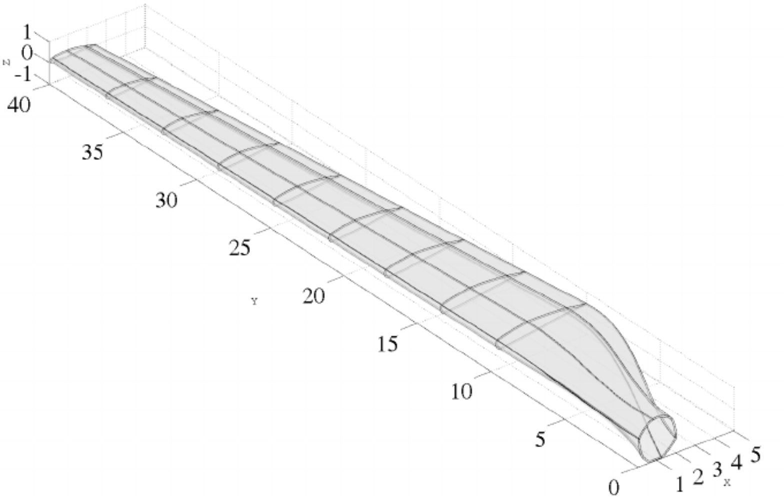
Department of Applied Mathematics, TU Delft, The Netherlands

Scientific machine learning workshop @ CWI, December 6-8, 2023

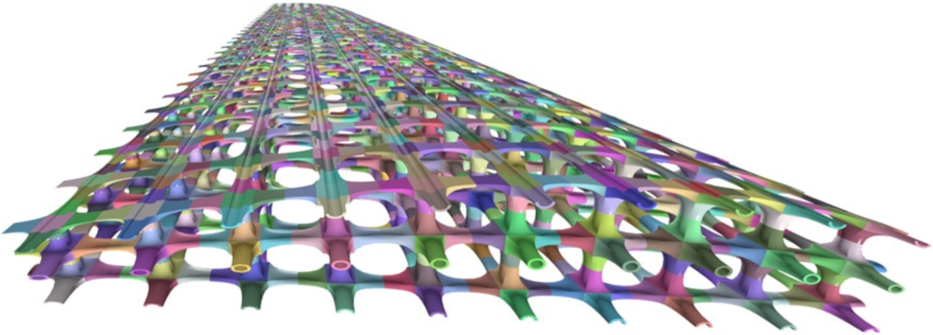
Design optimization: topology



Design optimization: shape



Design optimization: meso-/micro-structure and materials



How can SciML help?

- Create novel 'analysis-suitable' designs through generative AI

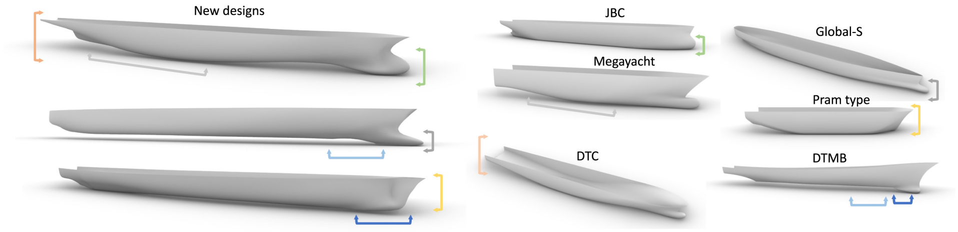


Figure from: *ShipHullGAN: A generic parametric modeller for ship hull design using deep convolutional generative model* [Khan et al., 2023]

How can SciML help?

- Create novel 'analysis-suitable' designs through generative AI
- Evaluate design's performance with physics-informed machine learning

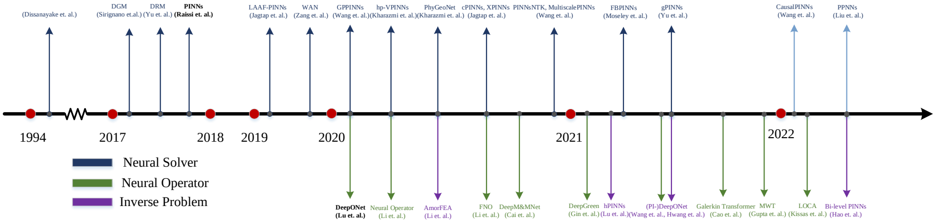


Figure from: *Physics-informed machine learning: A survey on problems, methods and applications* [Hao et al., 2022]

How can SciML help?

- Create novel ‘analysis-suitable’ designs through generative AI
- Evaluate design’s performance with physics-informed machine learning
- Speed up classical numerical methods with SciML technologies

How can SciML help?

- Create novel 'analysis-suitable' designs through generative AI
- Evaluate design's performance with physics-informed machine learning
- Speed up classical numerical methods with SciML technologies

How can SciML help?

- Create novel 'analysis-suitable' designs through generative AI
- Evaluate design's performance with physics-informed machine learning
- Speed up classical numerical methods with SciML technologies

Questions

1. How can we input 'analysis-suitable' designs into SciML?
2. How can we design a PIML framework that can be blended with classical methods?
   Aka: SciML for the quick pre-design analysis, <ABC> for in-depth analysis

How can SciML help?

- Create novel 'analysis-suitable' designs through generative AI
- Evaluate design's performance with physics-informed machine learning
- Speed up classical numerical methods with SciML technologies

Questions

1. How can we input 'analysis-suitable' designs into SciML?

   Our designs are described by B-spline/NURBS parametrizations

2. How can we design a PIML framework that can be blended with classical methods?
   Aka: SciML for the quick pre-design analysis, <ABC> for in-depth analysis

   *Our <ABC> is **Isogeometric Analysis** (IGA) [Hughes et al., 2005]*
   Aka: 'finite elements' based on B-spline/NURBS basis functions

Univariate B-splines

Knot vector

$$\Xi = \{\xi_1, \xi_2, \ldots, \xi_{n+d+1}\}, \qquad \xi_i \leq \xi_{i+1}, \quad \forall i = 1, \ldots, n+d$$

with ξ_i being a *knot*, n the *number* and d the *degree* of the B-spline basis functions

Recurrence formula [de Boor, 1971]

$$b_{i;\Xi}^{0}(\xi) = \begin{cases} 1 & \text{if } \xi_i \leq \xi < \xi_{i+1} \\ 0 & \text{otherwise} \end{cases}$$

$$b_{i;\Xi}^{d}(\xi) = \frac{\xi - \xi_i}{\xi_{i+d} - \xi_i} b_{i;\Xi}^{d-1}(\xi) + \frac{\xi_{i+d+1} - \xi}{\xi_{i+d+1} - \xi_{i+1}} b_{i+1;\Xi}^{d-1}(\xi) \qquad „\frac{0}{0}" := 0$$

Univariate B-spline properties

Local support and non-negativity

$$b_{i;\Xi}^{d}(\xi)\begin{cases} >0 & \forall\xi\in\text{supp}\left(b_{i;\Xi}^{d}\right):=[\xi_i,\xi_{i+d+1}),\\ =0 & \text{otherwise}\end{cases}$$

Partition of unity

$$\sum_{i=1}^{n} b_{i;\Xi}^{d}(\xi)\equiv 1,\quad \forall\xi\in\hat{I}_{\Xi}:=[\xi_1,\ldots\xi_{n+d+1})$$

Knot vectors

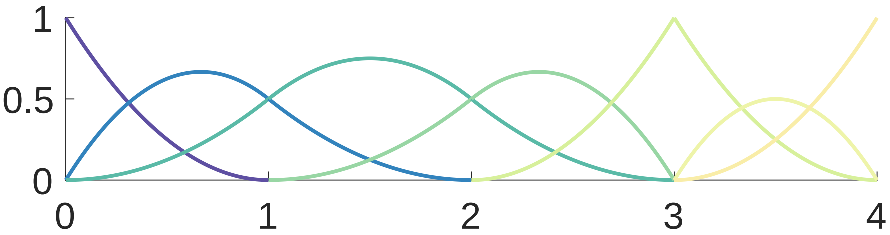
Open knot vector (i.e. $d+1$ repetition of the first and last knot)

$$\Xi = [\xi_1 = \cdots = \xi_{d+1}, \quad \xi_{d+2}, \ldots, \quad \xi_n = \cdots = \xi_{n+d+1}]$$

First and last basis functions are *interpolatory* at the left and right endpoint, respectively.

Repeated knots reduce the continuity of the basis functions that are non-zero at the respective knot from C^{d-1} to C^{d-m_i} *locally* with m_i being the multiplicity of the i-th knot.

The power of knot repetition



$$\Xi = \{0,0,0,1,2,3,3,4,4,4\}, \quad \hat{I}_\Xi = (0,4), \quad n = 7, \quad d = 2$$

The spline space $\mathbb{S}_{\Xi}^{d,s}$

$$\begin{aligned}\mathbb{S}_{\Xi}^{d,s} &= \text{span}\left\{b_{1;\Xi}^{d}, \dots, b_{n;\Xi}^{d}\right\} \\ &= \left\{\sum_{i=1}^{n} b_{i;\Xi}^{d}(\xi)\, c_i \;:\; c_i \in \mathbb{R}^s, \text{ for } 1 \le i \le n,\ \xi \in \hat{I}_{\Xi}\right\}\end{aligned}$$

Define spline function $f \in \mathbb{S}_{\Xi}^{d,s}$, i.e. mapping from $\hat{I}_{\Xi}$ to $\mathbb{R}^s$ through

$$f(\xi) = \begin{bmatrix} b_{1;\Xi}^{d}(\xi) & \dots & b_{n;\Xi}^{d}(\xi) \end{bmatrix} \cdot \begin{bmatrix} c_1 & \dots & c_n \end{bmatrix} = \mathbf{b} \cdot \mathbf{c}$$

and fix the B-spline coefficients $c_i \in \mathbb{R}^s$ relative to the given B-spline basis

The spline space $\mathbb{S}_{\Xi}^{d,s}$

$$
\begin{aligned}
\mathbb{S}_{\Xi}^{d,s} &= \text{span}\left\{b_{1;\Xi}^d, \dots, b_{n;\Xi}^d\right\} \\
&= \left\{\sum_{i=1}^{n} b_{i;\Xi}^d(\xi)\, c_i \;:\; c_i \in \mathbb{R}^s, \text{ for } 1 \le i \le n,\ \xi \in \hat{I}_{\Xi}\right\}
\end{aligned}
$$

Define spline function $f \in \mathbb{S}_{\Xi}^{d,s}$, i.e. mapping from $\hat{I}_{\Xi}$ to $\mathbb{R}^s$ through

$$
f(\xi) = \begin{bmatrix} b_{1;\Xi}^d(\xi) & \dots & b_{n;\Xi}^d(\xi) \end{bmatrix} \cdot \begin{bmatrix} c_1 & \dots & c_n \end{bmatrix} = \mathbf{b} \cdot \mathbf{c}
$$

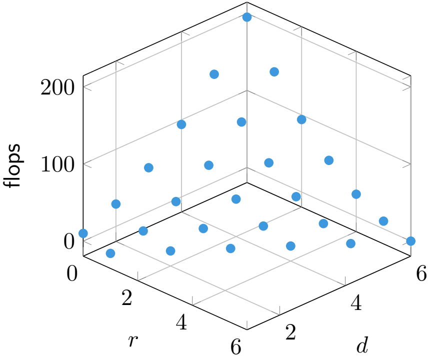
and fix the B-spline coefficients $c_i \in \mathbb{R}^s$ relative to the given B-spline basis

Extension to multiple dimensions via tensor-product construction $\rightarrow$ live demo

An efficient algorithm for evaluating univariate B-splines

Algorithm 2.22 from [Lyche and Mørken, 2018] with modifications

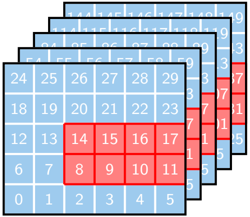
1. $\mathbf{b} = 1$
2. For $k = 1, \ldots, d-r$
   1. $\mathbf{t}_1 = (\xi_{i-k+1}, \ldots, \xi_i)$
   2. $\mathbf{t}_{21} = (\xi_{i+1}, \ldots, \xi_{i+k}) - \mathbf{t}_1$
   3. $\textsf{mask} = (\mathbf{t}_{21} < \textsf{tol})$
   4. $\mathbf{w} = (\xi - \mathbf{t}_1 - \textsf{mask}) \div (\mathbf{t}_{21} - \textsf{mask})$
   5. $\mathbf{b} = [(1-\mathbf{w}) \odot \mathbf{b}, 0] + [0, \mathbf{w} \odot \mathbf{b}]$
3. For $k = d-r+1, \ldots, d$
   1. $\mathbf{t}_1 = (\xi_{i-k+1}, \ldots, \xi_i)$
   2. $\mathbf{t}_{21} = (\xi_{i+1}, \ldots, \xi_{i+k}) - \mathbf{t}_1$
   3. $\textsf{mask} = (\mathbf{t}_{21} < \textsf{tol})$
   4. $\mathbf{w} = (1 - \textsf{mask}) \div (\mathbf{t}_{21} - \textsf{mask})$
   5. $\mathbf{b} = [-\mathbf{w} \odot \mathbf{b}, 0] + [0, \mathbf{w} \odot \mathbf{b}]$



where $\div$ and $\odot$ denote the element-wise division and multiplication of vectors, respectively.

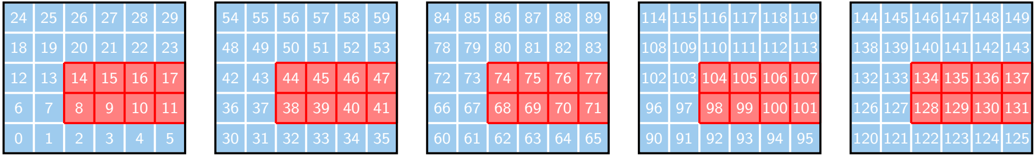
Memory layout of tensors

Example: $n_1 = 6, n_2 = 5, n_3 = 5$ and $d_1 = 3, d_2 = 1, d_3 = 4$



Memory layout of tensors

Example: $n_1 = 6, n_2 = 5, n_3 = 5$ and $d_1 = 3, d_2 = 1, d_3 = 4$



$$\mathbf{c} = [8 \quad 9 \quad 10 \quad 11 \quad 14 \quad 15 \quad 16 \quad 17 \quad 38 \quad 39 \quad \ldots \quad 136 \quad 137]$$

A brief recap of the Kronecker product

Definition

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} \mathbf{A}_{11}\mathbf{B} & \mathbf{A}_{12}\mathbf{B} \\ \mathbf{A}_{21}\mathbf{B} & \mathbf{A}_{22}\mathbf{B} \end{bmatrix} \qquad (n_A \cdot m_A \cdot n_B \cdot m_B \text{ flops})$$

Mixed-product property (if matrices are so that $\mathbf{AC}$ and $\mathbf{BD}$ exists)

$$(\mathbf{A} \otimes \mathbf{B})\,(\mathbf{C} \otimes \mathbf{D}) = (\mathbf{AC}) \otimes (\mathbf{BD})$$

Multiplicative decomposition (extendable to arbitrary number of matrices)

$$(\mathbf{A}_1 \otimes \mathbf{I}_2)\,(\mathbf{I}_1 \otimes \mathbf{A}_2) = (\mathbf{A}_1\mathbf{I}_1) \otimes (\mathbf{A}_2\mathbf{I}_2) = \mathbf{A}_1 \otimes \mathbf{A}_2$$

Efficient evaluation of multi-variate B-splines

It follows form the multiplicative decomposition of the Kronecker product that

$$f(\xi,\eta,\zeta) = \left(\mathbf{b}^{d_1} \otimes \mathbf{b}^{d_2} \otimes \mathbf{b}^{d_3}\right) \cdot \mathbf{c} = \left(\mathbf{I}_1 \otimes \mathbf{I}_2 \otimes \mathbf{b}^{d_3}\right) \cdot \left(\mathbf{I}_1 \otimes \mathbf{b}^{d_2} \otimes \mathbf{I}_3\right) \cdot \left(\mathbf{b}^{d_1} \otimes \mathbf{I}_2 \otimes \mathbf{I}_3\right) \cdot \mathbf{c}$$

Efficient evaluation of multi-variate B-splines

It follows form the multiplicative decomposition of the Kronecker product that

$$f(\xi,\eta,\zeta) = \left(\mathbf{b}^{d_1} \otimes \mathbf{b}^{d_2} \otimes \mathbf{b}^{d_3}\right) \cdot \mathbf{c} = \left(\mathbf{I}_1 \otimes \mathbf{I}_2 \otimes \mathbf{b}^{d_3}\right) \cdot \left(\mathbf{I}_1 \otimes \mathbf{b}^{d_2} \otimes \mathbf{I}_3\right) \cdot \left(\mathbf{b}^{d_1} \otimes \mathbf{I}_2 \otimes \mathbf{I}_3\right) \cdot \mathbf{c}$$

Algorithm 993 from [Fackler, 2019] with modifications

Set $\mathbf{f} := \mathbf{c}$

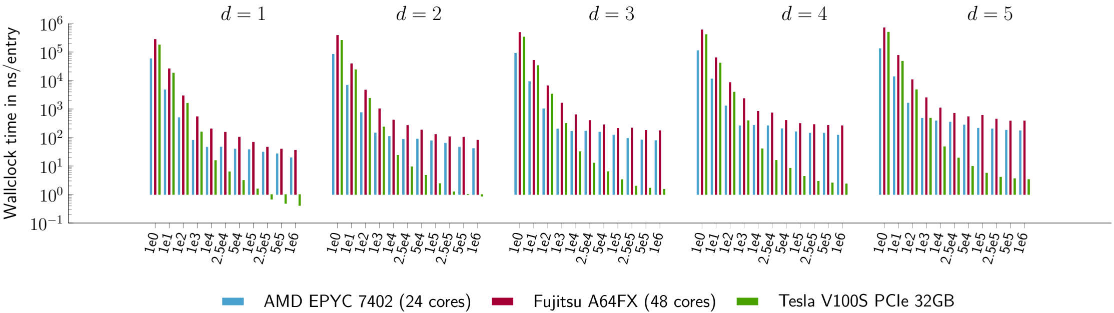
For $\ell = 1, 2, 3$

1. $\mathbf{f} :=$ reshape$(\mathbf{f}, [\cdot], d_\ell + 1)$
2. $\mathbf{f} := \mathbf{b}^{d_\ell} \cdot \mathbf{f}^\top$

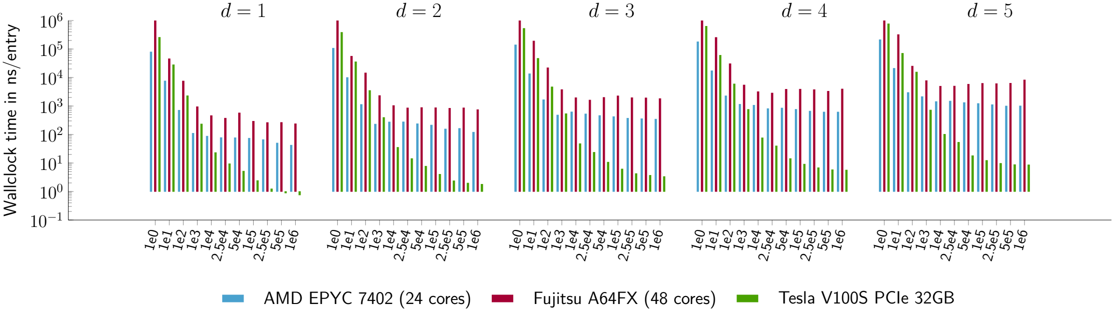
Output: $\mathbf{f} = f(\xi,\eta,\zeta)$

$$\mathbf{c} = \underbrace{\begin{bmatrix} 8 & 9 & 10 & 11 \\ 14 & 15 & 16 & 17 \\ 38 & 39 & 40 & 41 \\ \vdots & \vdots & \vdots & \vdots \\ 134 & 135 & 136 & 137 \end{bmatrix}}_{d_1+1 \text{ columns}} \begin{array}{c} (d_2+1)(d_3+1) \\ \text{rows} \end{array}$$

Performance evaluation - bivariate B-splines



Performance evaluation - trivariate B-splines



Collocation IGA

PDE problem

$$\mathcal{L}u = f \qquad \text{in } \Omega$$

$$\mathcal{B}u = g \qquad \text{on } \Gamma$$

Weighted residual form

$$\int_{\Omega} \phi_{\Omega}(\mathcal{L}u - f)\,\mathrm{d}\mathbf{x} + \int_{\Gamma} \phi_{\Gamma}(\mathcal{B}u - g)\,\mathrm{d}s = 0$$

Collocation IGA

PDE problem

$$\mathcal{L}u = f \qquad \text{in } \Omega$$

$$\mathcal{B}u = g \qquad \text{on } \Gamma$$

Weighted residual form

$$\int_\Omega \phi_\Omega(\mathcal{L}u - f)\,\mathrm{d}\mathbf{x} + \int_\Gamma \phi_\Gamma(\mathcal{B}u - g)\,\mathrm{d}s = 0$$

Let

$$\phi_\Omega = \sum_{i=1}^{k} \delta_\Omega(\mathbf{x} - \mathbf{x}_i)\,c_i \quad (\mathbf{x}_i \in \Omega) \qquad \text{and} \qquad \phi_\Gamma = \sum_{i=k+1}^{n} \delta_\Gamma(\mathbf{x} - \mathbf{x}_i)\,c_i \quad (\mathbf{x}_i \in \Gamma)$$

then

$$\sum_{i=1}^{k} \left(\mathcal{L}u(\mathbf{x}_i) - f(\mathbf{x}_i)\right)\,c_i + \sum_{i=1+k}^{n} \left(\mathcal{B}u(\mathbf{x}_i) - g(\mathbf{x}_i)\right)\,c_i = 0$$

Collocation IGA cont'd

As the coefficients c_i are arbitrary we obtain

$$\mathcal{L}u(\mathbf{x}_i) = f(\mathbf{x}_i) \qquad i = 1, \ldots, k$$

$$\mathcal{B}u(\mathbf{x}_i) = g(\mathbf{x}_i) \qquad i = k+1, \ldots, n$$

Collocation IGA cont'd

As the coefficients c_i are arbitrary and replacing $u \approx u_h = \sum_{j=1}^{n} b_j(\mathbf{x})u_j$ we obtain

$$\begin{bmatrix} \mathcal{L}b_1(\mathbf{x}_1) & \dots & \mathcal{L}b_n(\mathbf{x}_1) \\ \vdots & \ddots & \vdots \\ \mathcal{L}b_1(\mathbf{x}_k) & \dots & \mathcal{L}b_n(\mathbf{x}_k) \\ \mathcal{B}b_1(\mathbf{x}_{k+1}) & \dots & \mathcal{B}b_n(\mathbf{x}_{k+1}) \\ \vdots & \ddots & \vdots \\ \mathcal{B}b_1(\mathbf{x}_n) & \dots & \mathcal{B}b_n(\mathbf{x}_n) \end{bmatrix} \begin{bmatrix} u_1 \\ \vdots \\ u_k \\ u_{k+1} \\ \vdots \\ u_n \end{bmatrix} = \begin{bmatrix} f(\mathbf{x}_1) \\ \vdots \\ f(\mathbf{x}_k) \\ g(\mathbf{x}_{k+1}) \\ \vdots \\ g(\mathbf{x}_n) \end{bmatrix}$$

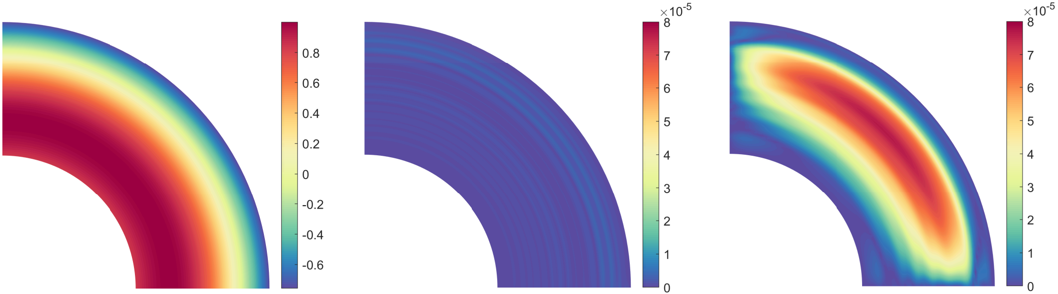
Collocation IGA cont'd

As the coefficients c_i are arbitrary and replacing $u \approx u_h = \sum_{j=1}^{n} b_j(\mathbf{x})u_j$ we obtain

$$\begin{bmatrix} \mathcal{L}b_1(\mathbf{x}_1) & \dots & \mathcal{L}b_n(\mathbf{x}_1) \\ \vdots & \ddots & \vdots \\ \mathcal{L}b_1(\mathbf{x}_k) & \dots & \mathcal{L}b_n(\mathbf{x}_k) \\ \mathcal{B}b_1(\mathbf{x}_{k+1}) & \dots & \mathcal{B}b_n(\mathbf{x}_{k+1}) \\ \vdots & \ddots & \vdots \\ \mathcal{B}b_1(\mathbf{x}_n) & \dots & \mathcal{B}b_n(\mathbf{x}_n) \end{bmatrix} \begin{bmatrix} u_1 \\ \vdots \\ u_k \\ u_{k+1} \\ \vdots \\ u_n \end{bmatrix} = \begin{bmatrix} f(\mathbf{x}_1) \\ \vdots \\ f(\mathbf{x}_k) \\ g(\mathbf{x}_{k+1}) \\ \vdots \\ g(\mathbf{x}_n) \end{bmatrix}$$

- basis functions b_i need to be at least C^ℓ such that $\mathcal{L}$ and $\mathcal{B}$ can be applied
- regular system matrix requires that #collocation points = #basis functions and all collocation points must be pairwise distinct

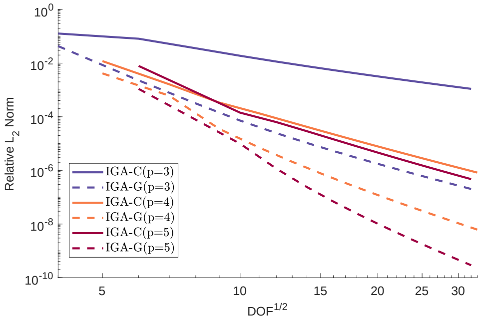
Comparison between Galerkin and collocation IGA



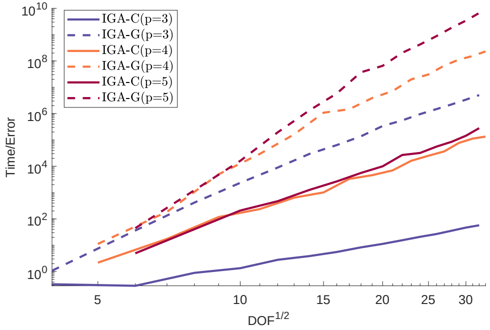
solution IGA-G abs. error IGA-C abs. error

Results by Mengyun Wang

Comparison between Galerkin and collocation IGA



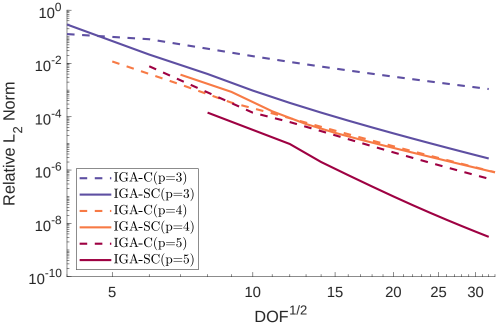
rel. L_2-error



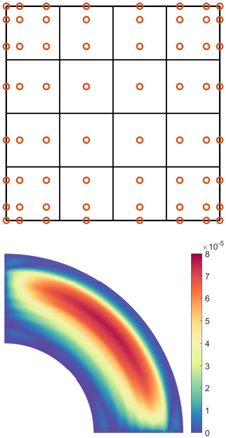
wallclock time/error

Results by Mengyun Wang

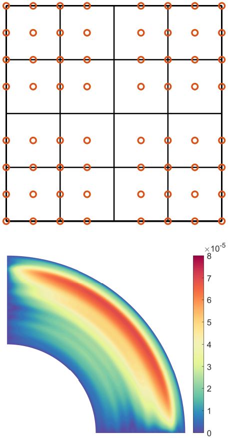
Comparison between Greville and clustered superconvergent points



rel. L_2-error



Greville



SC points

Results by Mengyun Wang

Least-squares collocation IGA

Idea: When #collocation points (m) $>$ #unknowns (n) then the system matrix is over-determined and the system can be solved in least-squares manner

$$\min_{u_h} \sum_{i=1}^{k} \|\mathcal{L}u_h(\mathbf{x}_i) - f(\mathbf{x}_i)\|^2 + \sum_{i=k+1}^{m} \|\mathcal{B}u_h(\mathbf{x}_i) - g(\mathbf{x}_i)\|^2$$

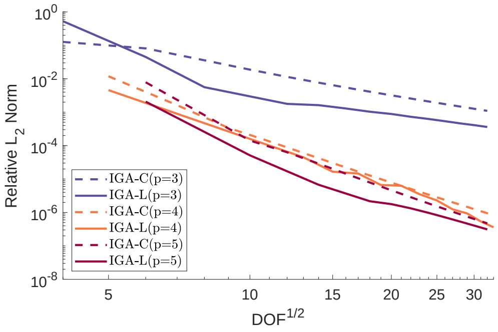
Least-squares collocation IGA

Idea: When #collocation points (m) $>$ #unknowns (n) then the system matrix is over-determined and the system can be solved in least-squares manner

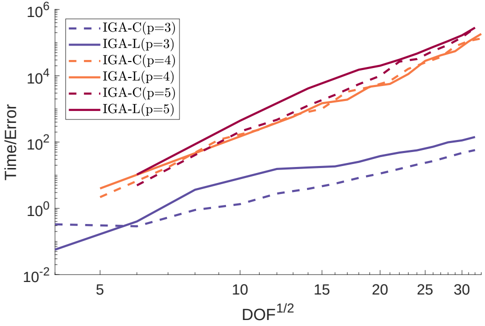
$$\min_{u_h} \sum_{i=1}^{k} \|\mathcal{L}u_h(\mathbf{x}_i) - f(\mathbf{x}_i)\|^2 + \sum_{i=k+1}^{m} \|\mathcal{B}u_h(\mathbf{x}_i) - g(\mathbf{x}_i)\|^2$$

[Lin et al., 2020] derives rigorous conditions under which least-squares collocation IGA (IGA-L) is consistent and convergent. In essence, there must be *at least one collocation point per element* (e.g., Greville points) but we can use more to increase the resolution.

Comparison between collocation and least-squares collocation IGA



rel. L_2-error



wallclock time/error

Results by Mengyun Wang

Least-squares collocation IGA revisited

Replacing f, and g by their approximations f_h, and g_h we obtain

$$\min_{\{u_j\}_j} \sum_{i=1}^{k} \| \sum_{j=1}^{n} \mathcal{L} b_j(\mathbf{x}_i) u_j - b_j(\mathbf{x}_i) f_j \|^2 + \sum_{i=k+1}^{m} \| \sum_{j=1}^{n} \mathcal{B} b_j(\mathbf{x}_i) u_j - b_j(\mathbf{x}_i) g_j \|^2$$

- B-spline basis functions $\hat{b}_j(\boldsymbol{\xi})$ are defined in the reference space $\hat{\Omega} = (0,1)^d$ and are mapped into physical space Ω through the **push-forward mapping**

$$\mathbf{x}_h(\boldsymbol{\xi}) = \sum_{i=1}^{n} \hat{b}_j(\boldsymbol{\xi}) \mathbf{x}_j,$$

Least-squares collocation IGA revisited

Replacing f, and g by their approximations f_h, and g_h we obtain

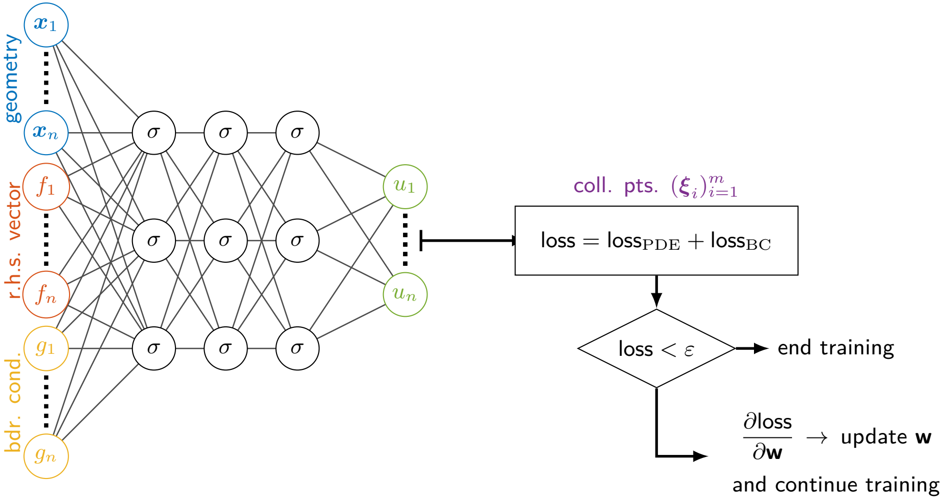
$$\min_{\{u_j\}_j} \underbrace{\sum_{i=1}^{k} \| \sum_{j=1}^{n} \mathcal{L} b_j(\mathbf{x}_i) u_j - b_j(\mathbf{x}_i) f_j \|^2}_{\mathsf{loss}_{\mathsf{PDE}}(\{u_j\}_j, \{f_j\}_j; \{\mathbf{x}_i\}_i)} + \underbrace{\sum_{i=k+1}^{m} \| \sum_{j=1}^{n} \mathcal{B} b_j(\mathbf{x}_i) u_j - b_j(\mathbf{x}_i) g_j \|^2}_{\mathsf{loss}_{\mathsf{BC}}(\{u_j\}_j, \{g_j\}_j; \{\mathbf{x}_i\}_i)}$$

- B-spline basis functions $\hat{b}_j(\boldsymbol{\xi})$ are defined in the reference space $\hat{\Omega} = (0,1)^d$ and are mapped into physical space Ω through the **push-forward mapping**

$$\mathbf{x}_h(\boldsymbol{\xi}) = \sum_{i=1}^{n} \hat{b}_j(\boldsymbol{\xi}) \mathbf{x}_j,$$

- problem is fully parameterized through f_j's, g_j's, and $\mathbf{x}_j$'s relative to a fixed basis $\hat{b}_j$

IgANet architecture



Training and evaluation

Training

For $[f_1, \dots, f_n] \in \mathcal{S}_{\text{rhs}}$, $[g_1, \dots, g_n] \in \mathcal{S}_{\text{bcond}}$, $[\mathbf{x}_1, \dots, \mathbf{x}_n] \in \mathcal{S}_{\text{geo}}$ **do**

For a batch of collocation points $\boldsymbol{\xi}_i \in [0, 1]^2$ (e.g., Greville points + more) **do**

Train IgANet $([f_1, \dots, f_n], [g_1, \dots, g_n], [\mathbf{x}_1, \dots, \mathbf{x}_n]) \mapsto [u_1, \dots, u_n]$

EndFor

EndFor

Training and evaluation

Training

For $[f_1, \dots, f_n] \in \mathcal{S}_{\text{rhs}}$, $[g_1, \dots, g_n] \in \mathcal{S}_{\text{bcond}}$, $[\mathbf{x}_1, \dots, \mathbf{x}_n] \in \mathcal{S}_{\text{geo}}$ **do**

For a batch of collocation points $\boldsymbol{\xi}_i \in [0, 1]^2$ (e.g., Greville points + more) **do**

Train IgANet $([f_1, \dots, f_n], [g_1, \dots, g_n], [\mathbf{x}_1, \dots, \mathbf{x}_n]) \mapsto [u_1, \dots, u_n]$

EndFor

EndFor

Evaluation

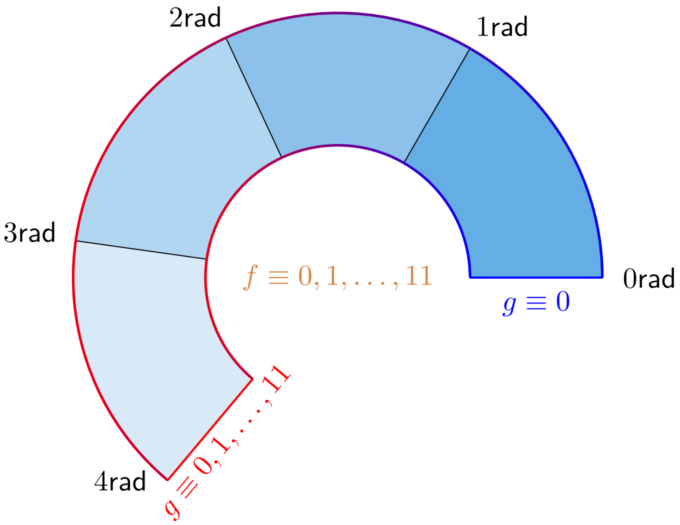
For $[f_1, \dots, f_n] \in \mathcal{S}_{\text{rhs}}$, $[g_1, \dots, g_n] \in \mathcal{S}_{\text{bcond}}$, $[\mathbf{x}_1, \dots, \mathbf{x}_n] \in \mathcal{S}_{\text{geo}}$ **do**

Evaluate IgANet $([f_1, \dots, f_n], [g_1, \dots, g_n], [\mathbf{x}_1, \dots, \mathbf{x}_n]) \mapsto [u_1, \dots, u_n]$

Use basis representation $u_h(\mathbf{x}) = \sum_{j=1}^{n} b_j(\mathbf{x}) u_j$ for all further purposes

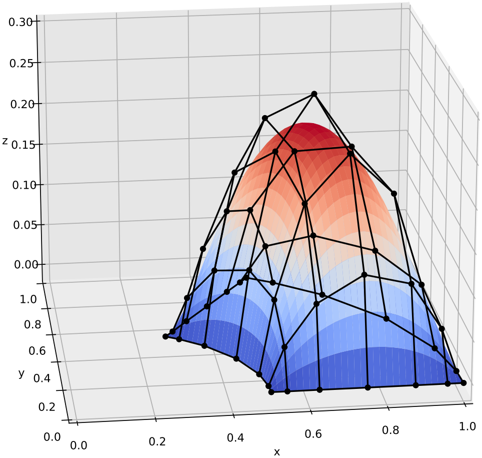
EndFor

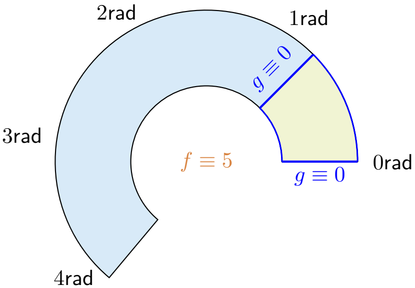
Test case: Poisson's equation on a variable annulus



Master thesis work by Frank van Ruiten, TU Delft

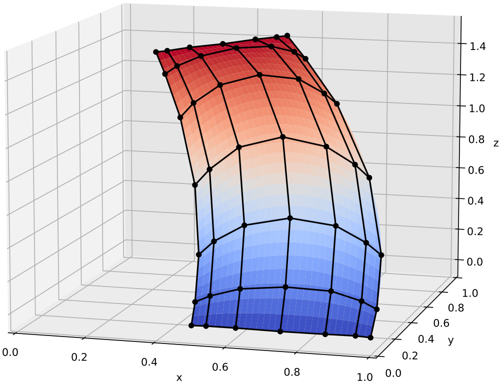
Validation results

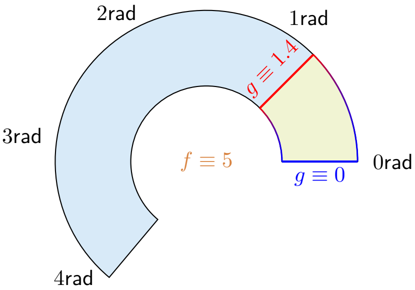




Master thesis work by Frank van Ruiten, TU Delft

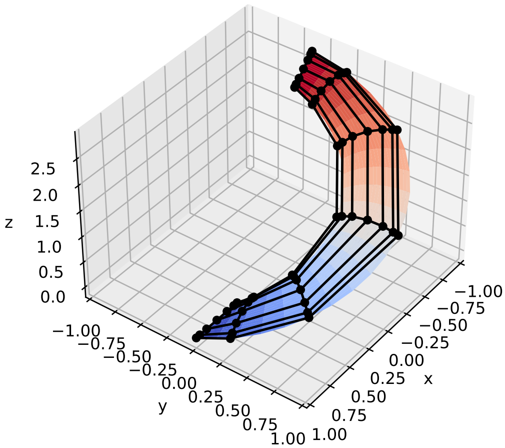
Validation results

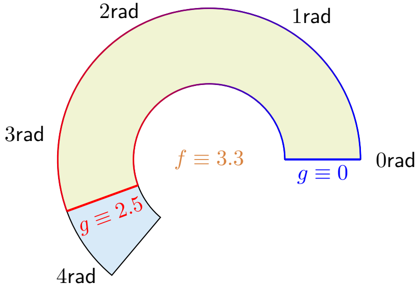




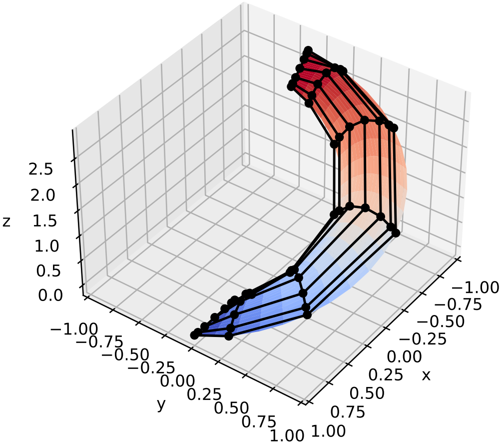
Master thesis work by Frank van Ruiten, TU Delft

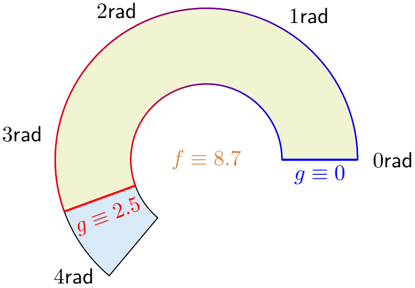
Validation results





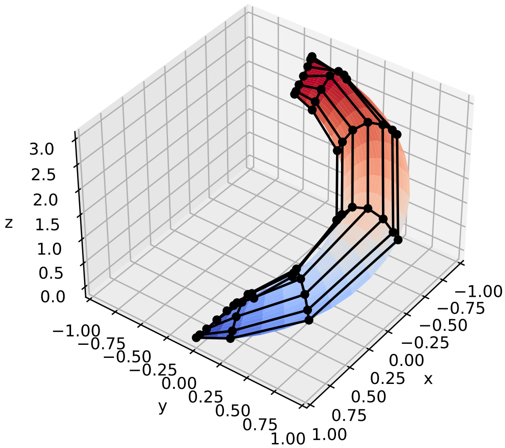
Validation results

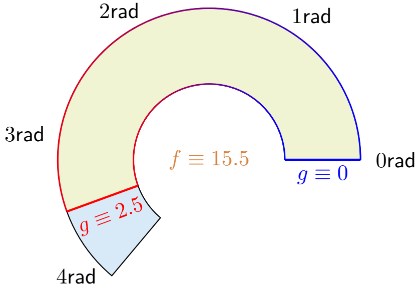




Master thesis work by Frank van Ruiten, TU Delft

Validation results





Master thesis work by Frank van Ruiten, TU Delft

Summary and outlook

- Least-squares collocation IGA enables seamless in-paradigm blending between fast learning-based pre-analysis and in-depth simulation-based (post-)analysis
- Theory from IGA-L carries over to NN (e.g., interpretation of loss function)
- Iterative refinement of NN's output by 'classical' IGA-L is possible

What's next?

- Interactive collaborative design modelling and optimization workflow

`https://visualization.surf.nl/iganet/`

Physics-informed machine learning in design optimization

Matthias Möller

Department of Applied Mathematics, TU Delft, The Netherlands

Scientific machine learning workshop @ CWI, December 6-8, 2023

Thank you very much!

References I

C. de Boor. Subroutine package for calculating with B-splines. (LA-4728), 1 1971. doi: 10.2172/4740859. URL https://www.osti.gov/biblio/4740859.

P. L. Fackler. Algorithm 993. *ACM Transactions on Mathematical Software*, 45(2):1–9, May 2019. doi: 10.1145/3291041. URL https://doi.org/10.1145/3291041.

Z. Hao, S. Liu, Y. Zhang, C. Ying, Y. Feng, H. Su, and J. Zhu. Physics-informed machine learning: A survey on problems, methods and applications, 2022. URL https://arxiv.org/abs/2211.08064.

T. Hughes, J. Cottrell, and Y. Bazilevs. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. *Computer Methods in Applied Mechanics and Engineering*, 194(39-41):4135–4195, Oct. 2005. doi: 10.1016/j.cma.2004.10.008. URL https://doi.org/10.1016/j.cma.2004.10.008.

References II

S. Khan, K. Goucher-Lambert, K. Kostas, and P. Kaklis. Shiphullgan: A generic parametric modeller for ship hull design using deep convolutional generative model. *Computer Methods in Applied Mechanics and Engineering*, 411:116051, 2023. ISSN 0045-7825. doi: https://doi.org/10.1016/j.cma.2023.116051. URL https://www.sciencedirect.com/science/article/pii/S0045782523001755.

H. Lin, Y. Xiong, X. Wang, Q. Hu, and J. Ren. Isogeometric least-squares collocation method with consistency and convergence analysis. *Journal of Systems Science and Complexity*, 33(5):1656–1693, Oct. 2020. doi: 10.1007/s11424-020-9052-9. URL https://doi.org/10.1007/s11424-020-9052-9.

T. Lyche and K. Mørken. Lecture notes: Spline methods, 2018. URL https://www.uio.no/studier/emner/matnat/math/MAT4170/v18/pensumliste/splinebook-2018.pdf.