Geometric shape optimization for Dirichlet energy with physics-informed and symplectic neural networks

A. Bélières–Frendo[1], E. Franck[2], V. Michel-Dansac[2], Y. Privat[3]

[1]IRMA, Université de Strasbourg, CNRS UMR 7501, 7 rue René Descartes, 67084 Strasbourg, France
[2]Université de Strasbourg, CNRS, Inria, IRMA, F-67000 Strasbourg, France
[3]Université de Lorraine, CNRS, Inria, IECL, F-54000 Nancy, France

IRMA

Université de Strasbourg

NumPEx

The authors acknowledge the financial support of the ExaMA project.

Introduction

- **The governing Partial Differential Equation (PDE)**. We study the Poisson equation

$$(\mathcal{P})\begin{cases}-\Delta u_\Omega^f = f & \text{in } \Omega,\\ u_\Omega^f = 0 & \text{on } \partial\Omega.\end{cases}$$

  - Ω is an open bounded connected set in $\mathbb{R}^n$
  - $f \in H^{-1}(\Omega)$ is the source term
  - u_Ω^f is the unique solution in $H_0^1(\Omega)$ of the Poisson problem

  The solution u_Ω^f of the Poisson problem $(\mathcal{P})$ is defined as the unique solution of the variational problem

$$(\mathcal{O}_\mathcal{P})\ \inf\{\mathcal{J}(\Omega,u),\ u \in H_0^1(\Omega)\},$$

$$\text{with } \mathcal{J}(\Omega,u) = \frac{1}{2}\int_\Omega |\nabla u|^2 - \langle f,u\rangle_{H^{-1}(\Omega),H_0^1(\Omega)}, \quad \forall u \in H_0^1(\Omega).$$

- **The shape optimization problem**. Introduce the Dirichlet energy $\mathcal{E}$, a shape functional given by

$$\mathcal{E}(\Omega) := \inf_{u\in H_0^1(\Omega)} \mathcal{J}(\Omega,u).$$

  Note that $\mathcal{E}(\Omega) = \mathcal{J}(\Omega,u_\Omega^f)$. Minimizing the Dirichlet energy within sets of given volum $V_0 > 0$ is a prototypal problem in shape optimization. It reads

$$(\mathcal{O}_\mathcal{D})\ \inf\{\mathcal{E}(\Omega),\ \Omega \text{ bounded set of } \mathbb{R}^n, \text{ such that } |\Omega| = V_0\}.$$

- **Objective**. Solve this problem with a *Neural Network (NN)*. We mention some of NNs advantages in the following non-exhaustive list.

  1. Automatic Differentiation (AD) avoids truncation errors;
  2. Parametric set of of source terms, or computational domains, thanks to Monte-Carlo integration;
  3. Mesh-free: work on very complex topologies
  4. Parallel code: joint gradient descent on several mutually dependent networks thanks to NNs. We train a network representing the solution of the PDE, and another network representing the computational domain.

1. PINNs and DeepRitz

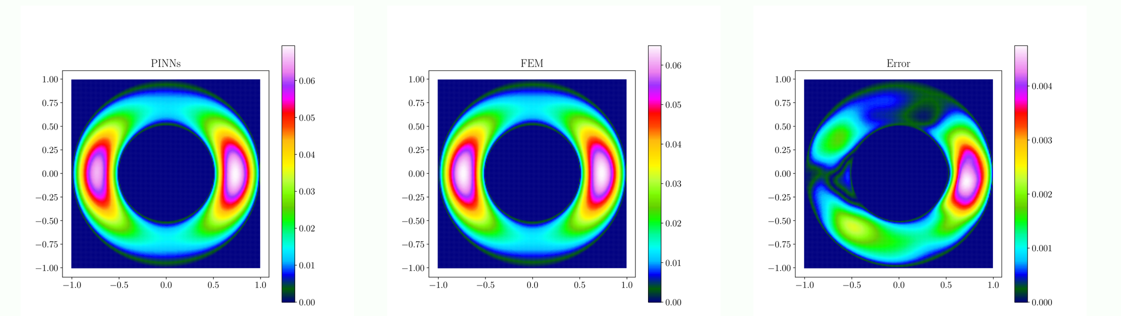
We want to solve the problem $(\mathcal{P})$ in a fixed shape with a PINN. For that, we minimize the following loss function [1]

$$\mathcal{J}_{PDE}(\theta) = \frac{V_0}{N}\sum_{i=1}^{N}\frac{1}{2}|\nabla v_\theta(x_i)|^2 - f(x_i)v(x_i),$$

- θ is the trainable set of parameters of the PINN;
- $v_\theta = \alpha u_\theta$ is the approximation of the solution of $(\mathcal{P})$;
- u_θ is the PINN;
- α is a C^∞ function, such that $\gamma_0^{\partial\Omega}\alpha = 0$ (for instance,if Ω is a disk, $\alpha(x,y) = 1 - x - y$);
- N is the number of the $\{x_i\}_{i=1}^N$ collocation points.

1.1. Numerical results

For the first simulation, in 2D, we solve the Poisson equation $(\mathcal{P})$ in an annulus, with the source term $f = \exp\left(1-(x/2)^2-(2y)^2\right)$. The 4 layers of the network have 10, 20, 20 and 10 neurons respectively and the learning rate is 5.10^{-3}. We compare it to a Finite Element Method (FEM) simulation done with FreeFem++ with a 500×500 mesh.



1.2. What about the parametric problem?

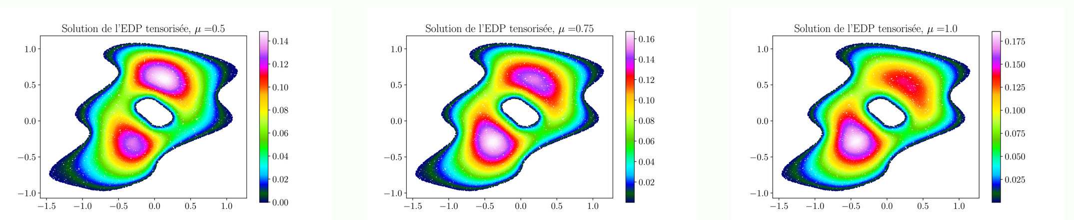
Consider a source term f^p which depends on parameters $\mu \in \mathbb{M}$. The parametric problem $(\mathcal{P}^p)$ can still be solved for a slightly larger but comparable computation time. The parametric Poisson problem then reads

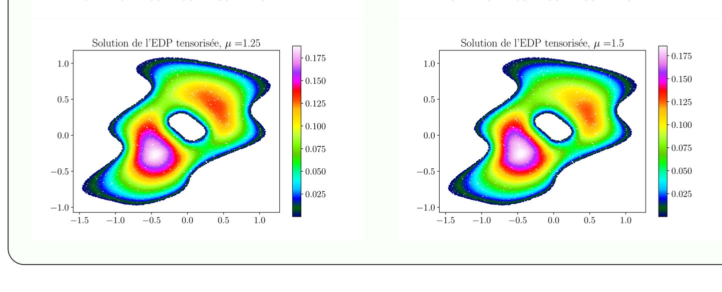
$$(\mathcal{P}^p)\begin{cases}-\Delta u^p(x;\mu) = f^p(x;\mu), & \text{for } (x;\mu) \in \Omega\times\mathbb{M};\\ u^p(x;\mu) = 0, & \text{for } (x;\mu) \in \partial\Omega\times\mathbb{M}.\end{cases}$$

- $\mathbb{M}$ is the space of parameters;
- $f^p : \Omega\times\mathbb{M} \to \mathbb{R}$ is the parametric source term.

We train the loss function $\mathcal{J}^p$ with $\{x_i,\mu_i\}_{i=1}^N \in \Omega\times\mathbb{M}$

$$\mathcal{J}_{PDE}^p\left(\theta;\{x_i,\mu_i\}_{i=1}^N\right) = \frac{V_0}{N}\sum_{i=1}^{N}\left\{\frac{1}{2}|\nabla v_\theta^p|^2 - f^p v_\theta^p\right\}(x_i;\mu_i),$$





Same simulation parameters in a potatoïd, $f(x,y;\mu) = \exp\left(1-(x/\mu)^2-(\mu y)^2\right)$.

2. Symplectic NNs (SympNets)

This work is concerned with **geometric optimization**. In $\mathbb{R}^2$, the **volume-preserving differentiable maps** are the **symplectic maps** (one can think of the flow of a Hamiltonian ordinary differential equation).

Objective: Train a symplectic NN to transform a disk into a given shape.

2.1. Principle of SympNets

- **Architecture**

Definition (shear maps): One of the simplest families of symplectic transformations from $\mathbb{R}^{2d}$ into $\mathbb{R}^{2d}$ is called "shear maps", and is defined as follows

$$f_{\text{up}}\begin{pmatrix}x\\y\end{pmatrix} = \begin{pmatrix}x+\nabla V_{\text{up}}(y)\\y\end{pmatrix};\ f_{\text{down}}\begin{pmatrix}x\\y\end{pmatrix} = \begin{pmatrix}x\\y+\nabla V_{\text{down}}(x)\end{pmatrix},$$

where $V_{\text{up/down}} \in C^1(\mathbb{R}^d,\mathbb{R})$, and $\nabla V : \mathbb{R}^d \to \mathbb{R}^d$ is the gradient of V.

Lemma [2]: Any symplectic map can be approximated by the composition of several shear maps.

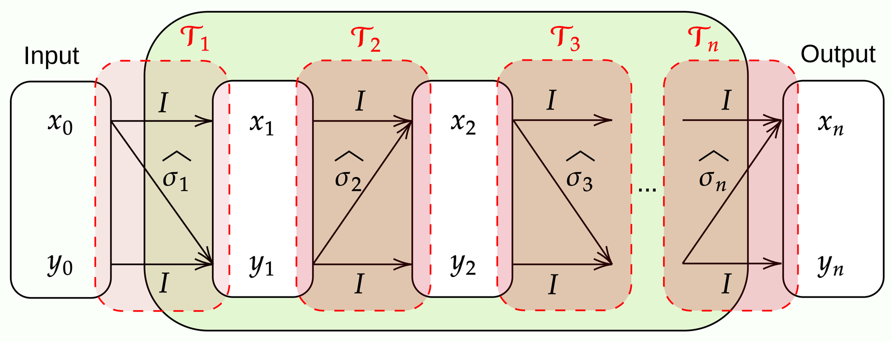
Theorem [2]: Let $q > 0$ be the depth of the NN, and $2d$ the dimension of the state space (here $d = 1$). We define $\widehat{\sigma_{K,a,b}}$ the approximation of ∇V in terms of an activation function σ, a vector $b \in \mathbb{R}^q$, a matrix $K \in \mathcal{M}_{q,2d}(\mathbb{R})$ and $a \in \mathbb{R}^q$ a vector, and $\text{diag}(a) = (a_i\delta_{ij})_{1\le i,j\le q}$, as follows

$$\widehat{\sigma_{K,a,b}}(x) = K^t\text{diag}(a)\sigma(Kx+b).$$

We define the gradient modules $\mathcal{G}_{\text{up}}$ and $\mathcal{G}_{\text{down}}$ to approximate f_{up} and f_{down}

$$\mathcal{G}_{\text{up}}\begin{pmatrix}x\\y\end{pmatrix} = \begin{pmatrix}x+\widehat{\sigma_{K,a,b}}(y)\\y\end{pmatrix};\quad \mathcal{G}_{\text{down}}\begin{pmatrix}x\\y\end{pmatrix} = \begin{pmatrix}x\\y+\widehat{\sigma_{K,a,b}}(x)\end{pmatrix}.$$

These functions are called gradient modules because $\widehat{\sigma_{K,a,b}}$ is able to approximate any ∇V.



- **Loss function**

To learn a given symplectic map $\mathcal{T}_{\text{objective}}$ with a SympNet $\mathcal{T}_\omega$, we minimize with respect to ω the following loss function [2]

$$\mathcal{J}_S(\omega,\{x_i\}_{i=1}^N) = \sum_{i=1}^{N}|\mathcal{T}_\omega(x_i) - \mathcal{T}_{\text{objective}}(x_i)|^2,$$

where ω are trainable weights and $\{x_i\}_{i=1}^N$ are N collocation points.

- **Can we make it parametric?**

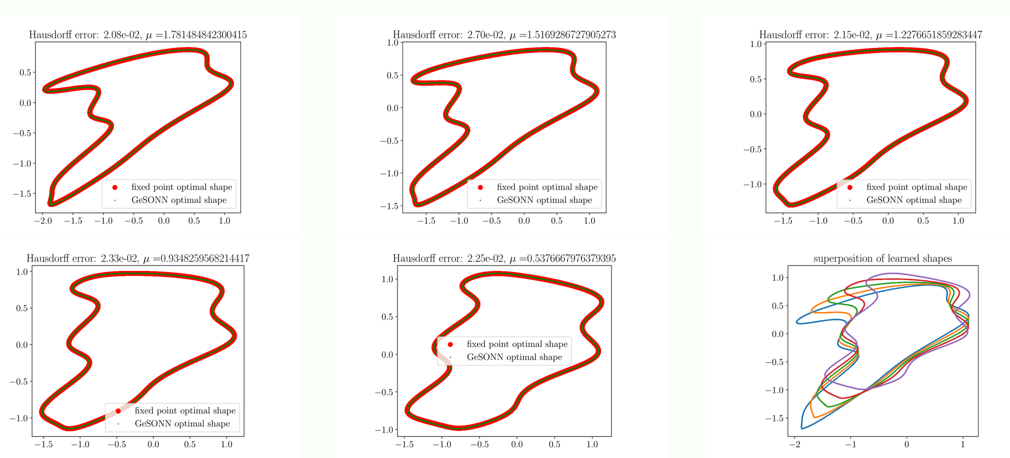
We propose to introduce $K_\mu \in \mathcal{M}_{q,n_\mu}(\mathbb{R})$ (with n_μ the number of parameters) and replace $\widehat{\sigma_{K,a,b}}(x)$ with [4]

$$\widetilde{\sigma}_{K,K_\mu,a,b}(x;\mu) = K^t\sigma(Kx+b+K_\mu\mu).$$

2.2. Numerical results

Here, we trained a SympNet to learn the parametric family of symplectic maps $\mathcal{T}_\mu = \mathcal{S}_\mu^1 \circ \mathcal{S}_\mu^2$, with

$$\begin{cases}\mathcal{S}_\mu^1 : (x,y;\mu) \mapsto (x-\mu y^2+0.3\sin(\frac{y}{\mu})-0.2\sin(8y),\quad y)\\ \mathcal{S}_\mu^2 : (x,y;\mu) \mapsto (x,\quad y+0.2\mu x+0.12\cos(x))\end{cases}$$



The SympNet was trained with 4 up and down networks, a width $q = 5$ and with a learning rate equal to 10^{-2}.

3. Shape optimization with NNs

3.1. Solving a PDE in a shape generated by a symplectic map

Objective: Train a SympNet and a PINN to transform a circle into the optimal shape for the Dirichlet energy.

Lemma [4]: Let $\mathcal{T}$ be a differentiable map and $u_\mathcal{T} \in H_0^1(\mathcal{T}\mathcal{C})$, such that $w = u_\mathcal{T}\circ\mathcal{T} \in H_0^1(\mathcal{C})$. If $u_\mathcal{T}$ is the solution of $(\mathcal{P})$, then w is solution of

$$(\mathcal{P}_\mathcal{T})\quad\begin{cases}-\text{div}(A\nabla w) = \widetilde{f}, & \text{in } \mathcal{C};\\ w = 0, & \text{on } \mathcal{C}.\end{cases}$$

1. $A : \mathcal{C} \to \mathbb{R} = J_\mathcal{T}^{-1}\cdot J_\mathcal{T}^{-t}$ is a uniformly elliptic metric tensor;
2. $J_\mathcal{T} = D\mathcal{T}$ the Jacobian matrix of $\mathcal{T}$ in the canonical basis of $\mathbb{R}^2$;
3. $\widetilde{f} = f\circ\mathcal{T} : \mathcal{C} \to \mathbb{R}$ the source term.

The previous problem can be formulated in a weaker sense, as the following optimization problem:

$$(\mathcal{O}_{\mathcal{P}_\mathcal{T}})\quad \inf\left\{\frac{1}{2}\int_\mathcal{C} A\nabla w\cdot\nabla w - \int_\mathcal{C}\widetilde{f}v,\quad w = u_\mathcal{T}\circ\mathcal{T} \in H_0^1(\mathcal{C})\right\}.$$

Remark: In spirit, we compute a shape derivative of $\mathcal{J}$.

3.2. Shape optimization loss function

We want to solve the problem $(\mathcal{O}_\mathcal{P})$. For that, we minimize the following parametric loss function [4]

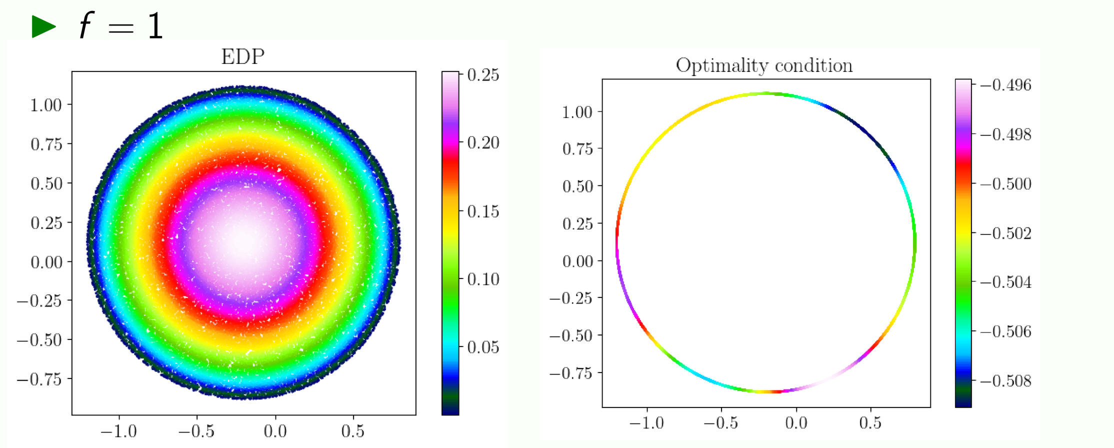
$$\mathcal{J}\left(\theta,\omega,\{(x_i;\mu_i)\}_{i=1}^N\right) = \frac{V_0}{N}\sum_{i=1}^{N}\left\{\frac{1}{2}|A_\omega\nabla v_{\theta,\omega}\cdot\nabla v_{\theta,\omega}|^2 - \widetilde{f}_\omega v_{\theta,\omega}\right\}(x_i;\mu_i)$$

- θ and ω the trainable weights for the PINN and the SympNet;
- $v_{\theta,\omega} : x \in \mathcal{C} \mapsto \alpha(x)u_\theta(\mathcal{T}_\omega x) + \beta(x) \in \mathbb{R}$ the solution of the Poisson problem set in $\mathcal{T}_\omega\mathcal{C}$;
- $\mathcal{T}_\omega : \mathbb{R}^{2d} \to \mathbb{R}^{2d}$ the SympNets;
- $A_\omega = J_{\mathcal{T}_\omega}^{-1}\cdot J_{\mathcal{T}_\omega}^{-t}$ the metric tensor;
- $u_\theta : \mathcal{T}_\omega\mathcal{C} \to \mathbb{R}$ the PINN;
- $\alpha : \mathcal{C} \mapsto \mathbb{R}$ a C^∞ function that vanishes on $\partial\mathcal{C}$;
- $\widetilde{f}_\omega : x \in \mathcal{C} \mapsto (f\circ\mathcal{T}_\omega)(x) \in \mathbb{R}$.

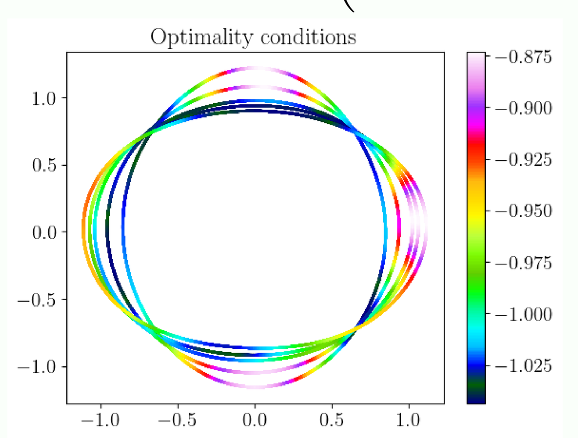
3.3. Numerical results

Theorem [3]: The problem $(\mathcal{O}_\mathcal{D})$ has a unique solution (Ω^*,u^*). The first order optimality condition reads: $\nabla u^*\cdot n$ is constant a.e. on $\partial\Omega^*$. For the numerical simulation, we take standard NNs settings in this section.

- $f = 1$



- $f(x,y;\mu) = \exp\left(1-(x/\mu)^2-(\mu y)^2\right)$

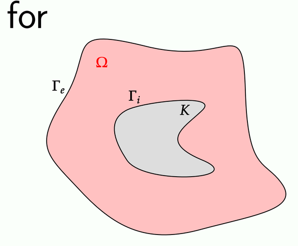


In this case, we simultaneously plot the optimality conditions for $\mathbb{M} = [0.5, 1.5]$.

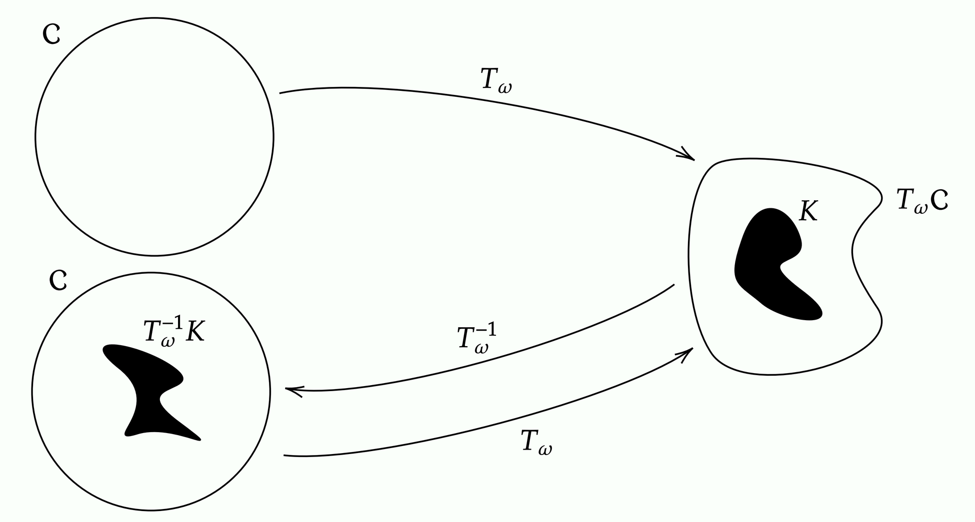
3.4. Ongoing work: the Bernoulli overdetermined problem

We now minimize the Dirichlet energy for

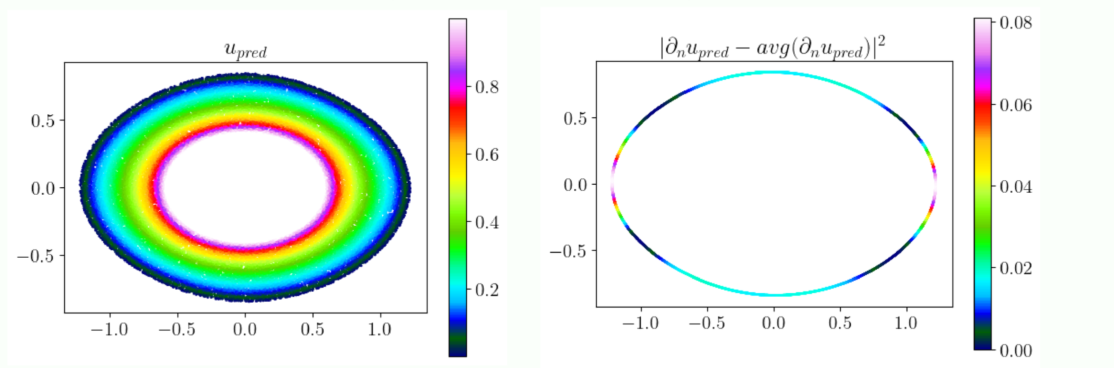
$$(\mathcal{B})\quad\begin{cases}-\Delta u = 0 & \text{in } \Omega;\\ u = 1 & \text{on } \Gamma_i;\\ u = 0 & \text{on } \Gamma_e.\end{cases}$$



Our numerical strategy remains to learn a symplectic map $\mathcal{T}_\omega$ minimizing the Dirichlet energy. To handle the boundary $\partial\Gamma_i$, we compute $\mathcal{T}_\omega^{-1}K$ [4].



For the numerical simulations, we take standard NNs settings, and K is an ellipse of parameters $(a = 0.6, b = 1/0.6)$.



Ongoing work and perspectives

- Publish the open source code GeSONN (GEometric Shape Optimization with Neural Networks)
- Investigate GeSONN for the compliance loss function
- Adapt GeSONN to the other equations
- Fixed point algorithm

References

[1] W. E, Y. Bing. The Deep Ritz method: A deep learning-based numerical algorithm for solving variational problems. *Commun. Math. Stat.* 6:1–12, (2018).

[2] P. Jin, Z. Zhang, A. Zhu, Y. Tang and G. E. Karniadakis. SympNets: Intrinsic structure-preserving symplectic networks for identifying Hamiltonian systems. *Neural Networks*, 132:166–179, 2020.

[3] A. Henrot and M. Pierre. **Shape Variation and Optimization: Geometrical Analysis.** *Mathématiques & Applications*, 2005.

[4] A. Bélières–Frendo, E. Franck, V. Michel-Dansac and Y. Privat. Geometric shape optimization for Dirichlet energy with NNs. *in preparation, 2024.*