

# How to work with complex geometries in PINNs ?

**Authors:**

Frédérique LECOURTIER

**Supervisors:**

Emmanuel FRANCK

Michel DUPREZ

Vanessa LLERAS

March 26, 2024

# Problem considered

**Poisson problem with Dirichlet conditions :**

Find  $u : \Omega \rightarrow \mathbb{R}^d$  ( $d = 1, 2, 3$ ) such that

$$\begin{cases} -\Delta u(X) = f(X) & \text{in } \Omega, \\ u(X) = g(X) & \text{on } \partial\Omega \end{cases}$$

with  $\Delta$  the Laplace operator,  $\Omega$  a smooth bounded open set and  $\Gamma$  its boundary.

For the following examples, we will consider  $f(X) = 1$  and  $g(X) = 0$ .



**Standard PINNs :** We are looking for  $\theta_u$  such that

$$\theta_u = \operatorname{argmin}_{\theta} w_r J_r(\theta) + w_{bc} J_{bc}(\theta)$$

where  $w_r$  and  $w_{bc}$  are the respective weights associated with

$$J_r = \int_{\Omega} (\Delta u + f)^2 \quad \text{and} \quad J_{bc} = \int_{\partial\Omega} (u - g)^2.$$

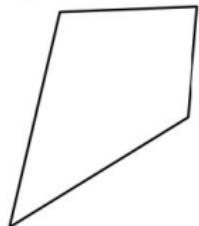
*Remark :* In practice, we use a Monte-Carlo method to discretize the cost function by random process.

# Simple geometry

**Claim on PINNs :** No mesh, so easy to go on complex geometry !

## Easy-to-sample shape

Quadrilateral



Cylinder

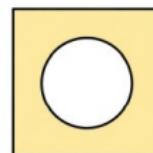


Ellipse

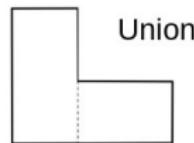


## Shape composition

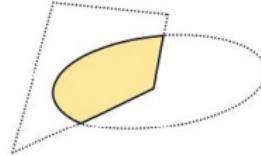
Subtraction



Union



Intersection



**In practice :** Not so easy ! We need to find **how to sample in the geometry**.

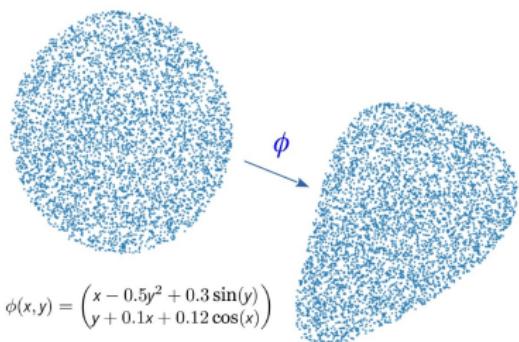
# Complex geometry

## 1st approach : Mapping

Idea :

- $\Omega_0$  a simple domain (as circle)
- $\Omega$  a target domain
- A mapping from  $\Omega_0$  to  $\Omega$  :

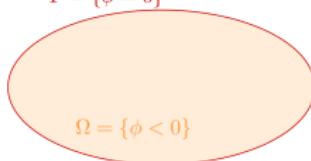
$$\Omega = \phi(\Omega_0)$$



## 2nd approach : LevelSet function

$$\Gamma = \{\phi = 0\}$$

$$\phi > 0$$



Advantages :

- Sample is easy in this case.
- Allow to impose hard the BC :  
$$u_\theta(X) = \phi(X)w_\theta(X) + g(X)$$

Natural LevelSet :

Signed Distance Function (SDF)

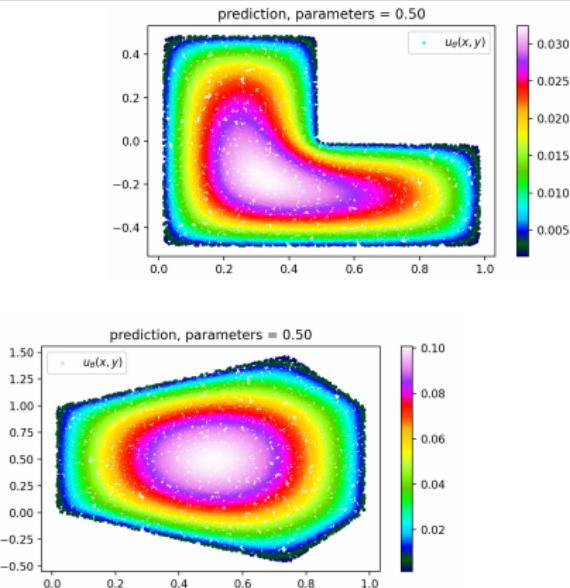
Problem : SDF is a  $C^0$  function  
⇒ its derivatives explodes  
⇒ we need a regular levelset

# Construct smooth SDF I

1st solution : Approximation theory [5]

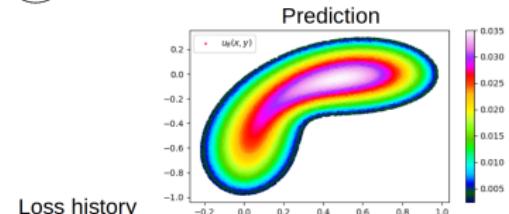
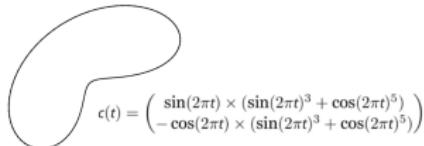
$\Delta\phi$  can be singular at the boundary. Sampling at  $\epsilon$  to it solve the problem.

Polygonal domain Appendix 1



Curved domain Appendix 2

Minus : Use of a parametric curve  $c(t)$ .



# Construct smooth SDF II

**2nd solution :** Learn the levelset. [2]

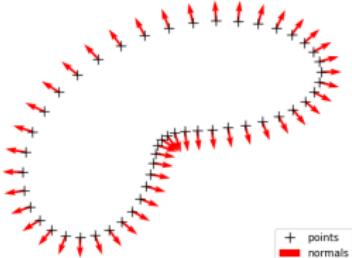
→ How make that ? with a PINNs.

If we have a boundary domain  $\Gamma$ , the SDF is solution to the Eikonal equation:

$$\begin{cases} \|\nabla\phi(x)\| = 1, & x \in \mathcal{O} \\ \phi(x) = 0, & x \in \Gamma \\ \nabla\phi(x) = n, & x \in \Gamma \end{cases}$$

with  $\mathcal{O}$  a box which contains  $\Omega$  completely and  $n$  the exterior normal to  $\Gamma$ .

**Advantage :** No need for parametric curves.



- set of boundary points
- exterior normals at  $\Gamma$   
(evaluated at these points)

# Learn LevelSet I

## Objective of the paper :

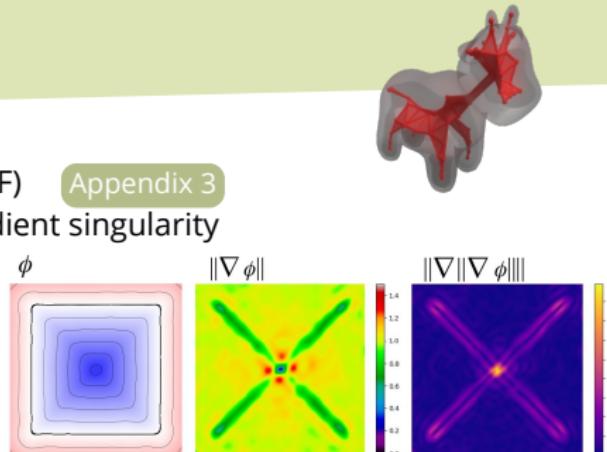
Learn topological Skeleton (by learning SDF) Appendix 3

→ Skeleton correspond exactly to the gradient singularity

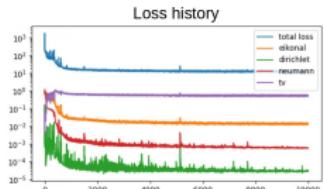
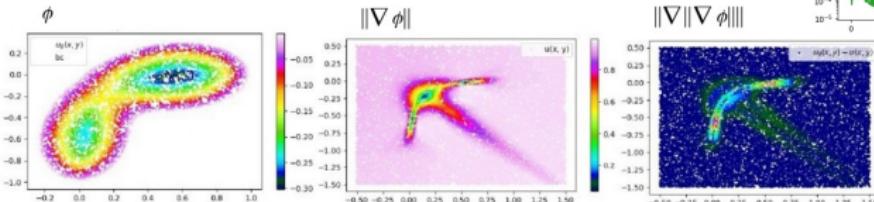
→ Adding the following term in the loss

$$\int_{\mathcal{O}} \|\nabla \|\nabla \phi\|\|(p)\| dp$$

(Total Variation Regularization)

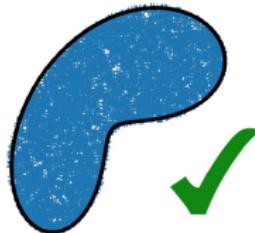


**1st test :** Eikonal equation with TV Regularization [2]

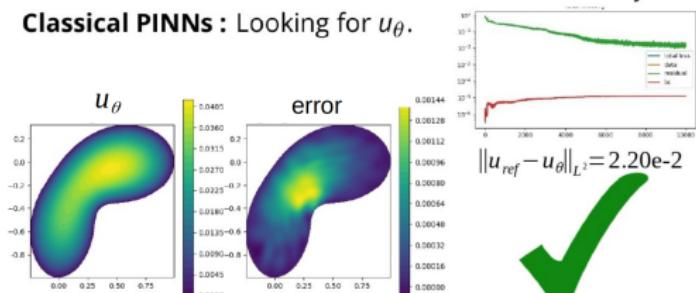


# Learn LevelSet I

Sampling :

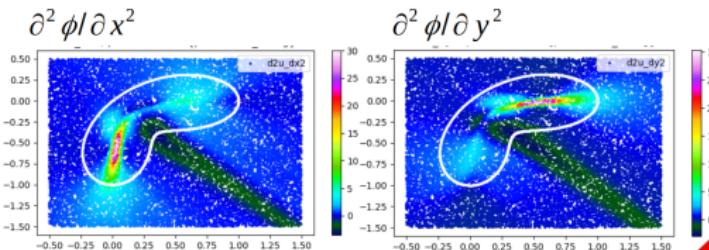


Classical PINNs : Looking for  $u_\theta$ .



Minus : Costly boundary points generation.

PINNs - Impose BC in hard : Looking for  $u_\theta = \phi w_\theta$ .

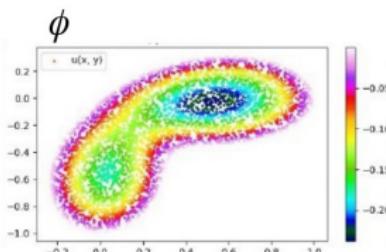


Levelset derivatives explode.

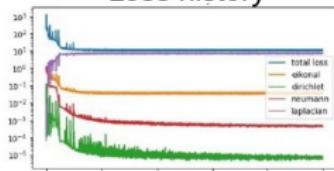
# Learn LevelSet II

**2nd test :** We replace the TV term by a penalization on the laplacian of the levelset

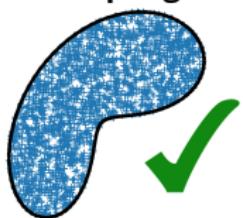
$$J_{reg} = \int_{\mathcal{O}} |\Delta \phi|^2$$



Loss history

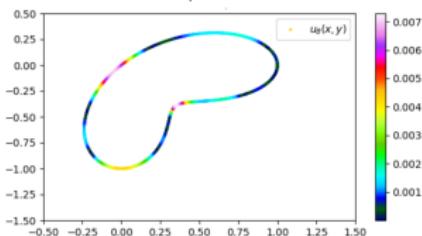


**Sampling :**



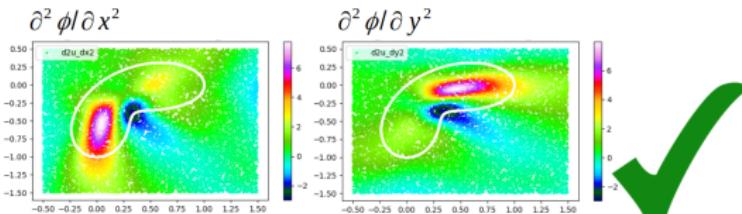
**Dirichlet error on the boundary :** Looking for  $u_\theta = \phi w_\theta$ .

Max : 7.29e-3 ; Mean : 1.88e-3



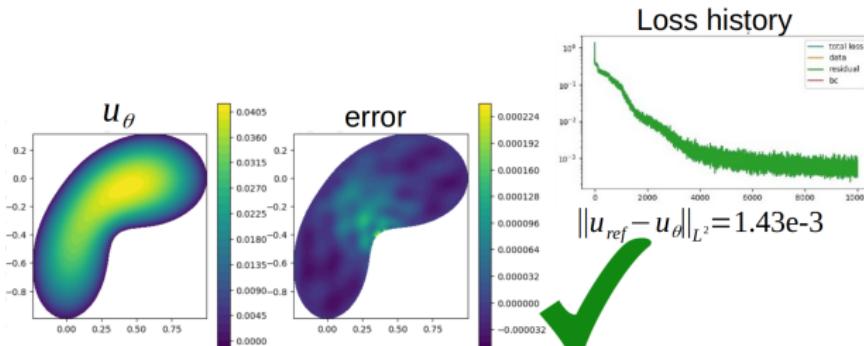
# Learn LevelSet II

Derivatives :



⇒ We can impose in hard boundary conditions

PINNs - Impose BC in hard :



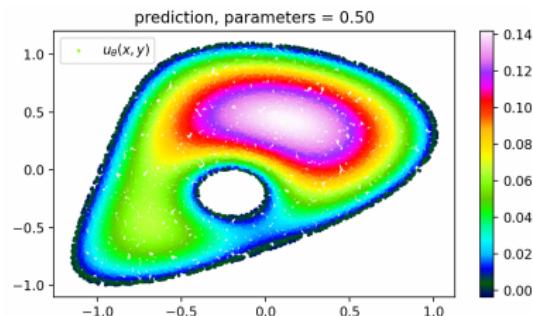
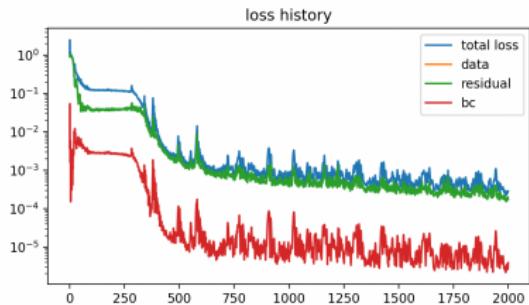
# Conclusion

## 2 main questions :

- How to sample in complex domains?
  - Using mapping
  - Using Levelset (Approximation theory/Learning)
- How can we obtain a levelset that usable for imposing boundary conditions in hard ?  
By learning the Eikonal equation with penalisation of the levelset Laplacian

**To go further :** We can combine the option.

(Mapping for the big domain. Level set for the hole.)



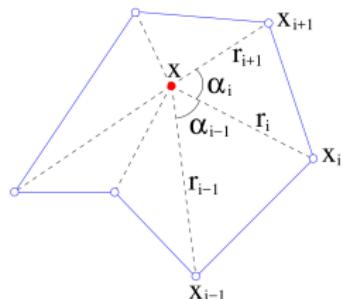
Thank you !

# Bibliography

- [1] Alexander Belyaev, Pierre-Alain Fayolle, and Alexander Pasko. Signed L<sub>p</sub>-distance fields. *Computer-Aided Design*.
- [2] Mattéo Clémot and Julie Digne. Neural skeleton: Implicit neural representation away from the surface. *Computers and Graphics*.
- [3] Pierre-Alain Fayolle. Signed Distance Function Computation from an Implicit Surface.
- [4] M. Raissi, P. Perdikaris, and G. E. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*.
- [5] N. Sukumar and Ankit Srivastava. Exact imposition of boundary conditions with distance functions in physics-informed deep neural networks. *Computer Methods in Applied Mechanics and Engineering*.
- [6] Sifan Wang, Shyam Sankaran, Hanwen Wang, and Paris Perdikaris. An Expert's Guide to Training Physics-informed Neural Networks.

# Appendix 1 : Polygonal domain [5]

- $X_i, i = 1, \dots, n$  - coordinates of the polygon
- $\alpha_i$  - angle between  $X_i$  and  $X_{i+1}$
- $r_i = \|X_i - X\|$  - Euclidean distance between  $X_i$  and  $X$
- $R_i = X_i - X$



We definte the SDF as

$$\phi(X) = \frac{2}{W(X)}$$

with

$$W(X) = \sum_{i=1}^n \left( \frac{1}{r_i} + \frac{1}{r_{i+1}} \right) t_i \quad (r_{n+1} := r_1)$$

and

$$t_i := \tan \left( \frac{\alpha_i}{2} \right) = \frac{\det(R_i, R_{i+1})}{r_i r_{i+1} + R_i \cdot R_{i+1}}$$

*Remark :* The denominator vanishes when  $\alpha_i = \pi$ , (i.e. when  $X$  lies on the boundary of the polygon), but there  $\phi_i(X) = 0$ .

## Appendix 2 : Curved domain [5]

Considering a nonconvex domain.

- $c(t)$  - parametrization of the curved boundary  $\Gamma : [0, 1] \rightarrow \mathbb{R}$
- $c'(t)$  - its tangent
- $c'^\perp(t)$  - rotating  $c'(t)$  through  $90^\circ$  (clockwise)

We definte the SDF as

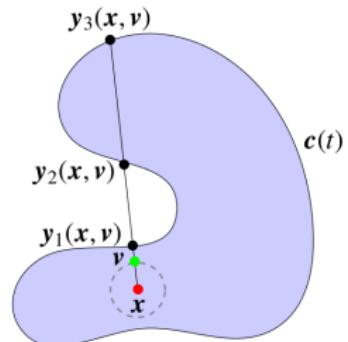
$$\phi(X) = \left( \frac{1}{W_p(X)} \right)^{1/p}$$

with

$$W_p(X) = \int_0^1 \frac{(c(t) - X) \cdot c'^\perp(t)}{\|c(t) - X\|^{2+p}}$$

(Belyaev et al. [1] introduced  $L_p$ -distance fields ( $p \geq 1$ ), which approximates the exact distance function.)

*Remark :* For  $X \in \Gamma$  (integral is singular), we set  $\phi(X) = 0$ .



# Appendix 3 : Neural Skeleton

**Simple example :** Skeleton of the unit square.

