



**HAL**  
open science

## **DS-GPS: A Deep Statistical Graph Poisson Solver**

Matthieu Nastorg, Marc Schoenauer, Guillaume Charpiat, Thibault Faney,  
Jean-Marc Gratién, Michele Alessandro Bucci

► **To cite this version:**

Matthieu Nastorg, Marc Schoenauer, Guillaume Charpiat, Thibault Faney, Jean-Marc Gratién, et al.. DS-GPS: A Deep Statistical Graph Poisson Solver. NeurIPS 2022 - Machine Learning and the Physical Sciences, workshop, Dec 2022, New-Orleans, United States. hal-03864015

**HAL Id: hal-03864015**

**<https://inria.hal.science/hal-03864015v1>**

Submitted on 21 Nov 2022

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

### Motivations

- Numerical simulations of fluid flow plays an essential role in modelling many physical phenomena but solving the incompressible Navier-Stokes equations at scale remains daunting.
- The splitting scheme method<sup>[1]</sup> requires the costly resolution of a Poisson problem to compute the pressure field that guarantees the incompressibility constraint.
- Leverage data-driven methods<sup>[2]</sup> to enhance the Poisson pressure problem (PPE) resolution.

### Statistical Problem

#### Continuous Poisson Problem

$$\begin{cases} -\Delta u_p = f_p & \in \Omega_p \\ u_p = g_p & \in \partial\Omega_p \end{cases}$$

Finite Element

#### Discretized Poisson Problem

$$A_p U_p = B_p$$

#### Define a loss function

$$\mathcal{L}_{\text{res}}(U_p, A_p, B_p) = \|A_p U_p - B_p\|^2$$

#### Machine-Learning solver

$$U_p = \text{solver}_{\theta}(A_p, B_p)$$

#### Statistical Problem

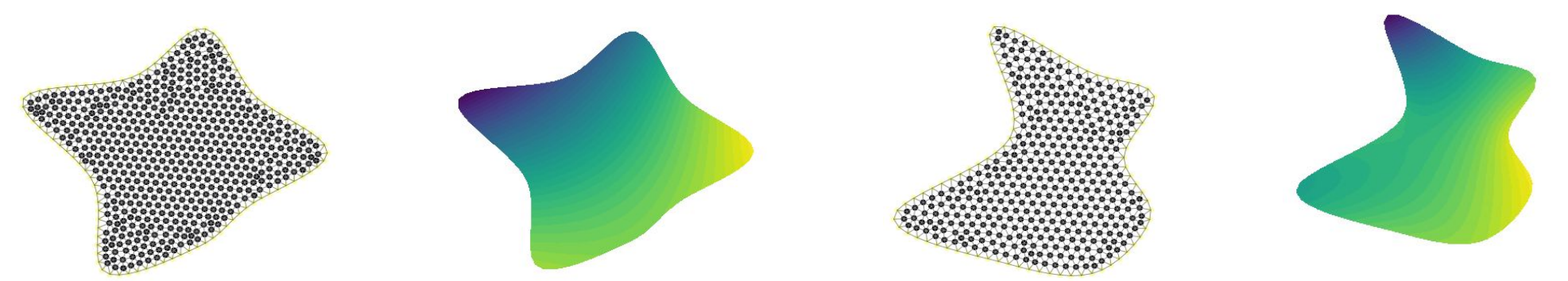
$$\theta^* = \underset{\theta \in \Theta}{\text{argmin}} \mathbb{E}_{p \sim \mathcal{D}} [\mathcal{L}_{\text{res}}(\text{solver}_{\theta}(U_p, A_p, B_p), A_p, B_p)]$$

### Objectives

- Develop a GNN-based algorithm to **speed-up** the resolution of the PPE on **unstructured meshes**.
- Design a model **flexible** to different **domains**, **boundary conditions** and **initial solutions**.
- The model should **respect the boundary conditions** by design and provide **convergence guarantees**.

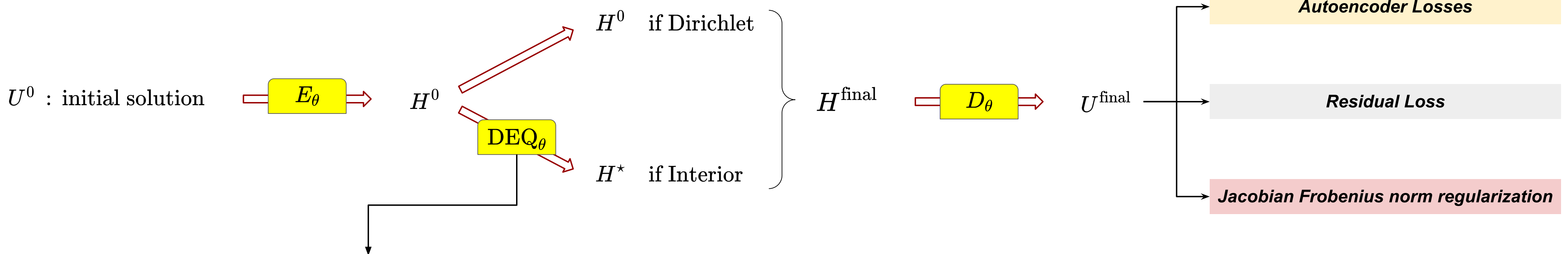
### The Data

- Random 2D domains with coordinates in the unit square.
- ≈ 500 nodes per geometries (Minimum : 221 - Maximum : 713).
- Force and boundary functions are defined as polynomials with randomly sampled coefficients.



Two different discretized domains (triangular mesh) with their corresponding solutions

### Architecture



### Deep Equilibrium<sup>[3]</sup> GNNs

- Use a **black-box solver** to find the root of the GNN function :

$$H^* = \text{RootFind}(g_{\theta}, x) \quad \text{with} \quad g_{\theta} = f_{\theta}(H, \mathcal{G}) - H$$

- The GNN function incorporates the different **Message Passing** :

$$\phi_{\rightarrow, i} = \sum_{j \in \mathcal{N}(i)} \Phi_{\rightarrow}(H_i, H_j, e_{ij})$$

- The number of Message Passing is **automatically determined** by the "RootFind" black-box solver (a quasi-Newton method) !

### Conclusion

This paper proposes a novel GNN-based approach to solve a Poisson problem with Dirichlet boundary conditions. Leveraging the implicit layer theory and a node-specific architecture, our method extends to any mesh size and shape, provides convergence guarantees and satisfies boundary conditions by design. By directly minimizing the residual of the problem, the model learns the physics of the equation and can output a solution ten times faster than traditional solvers.

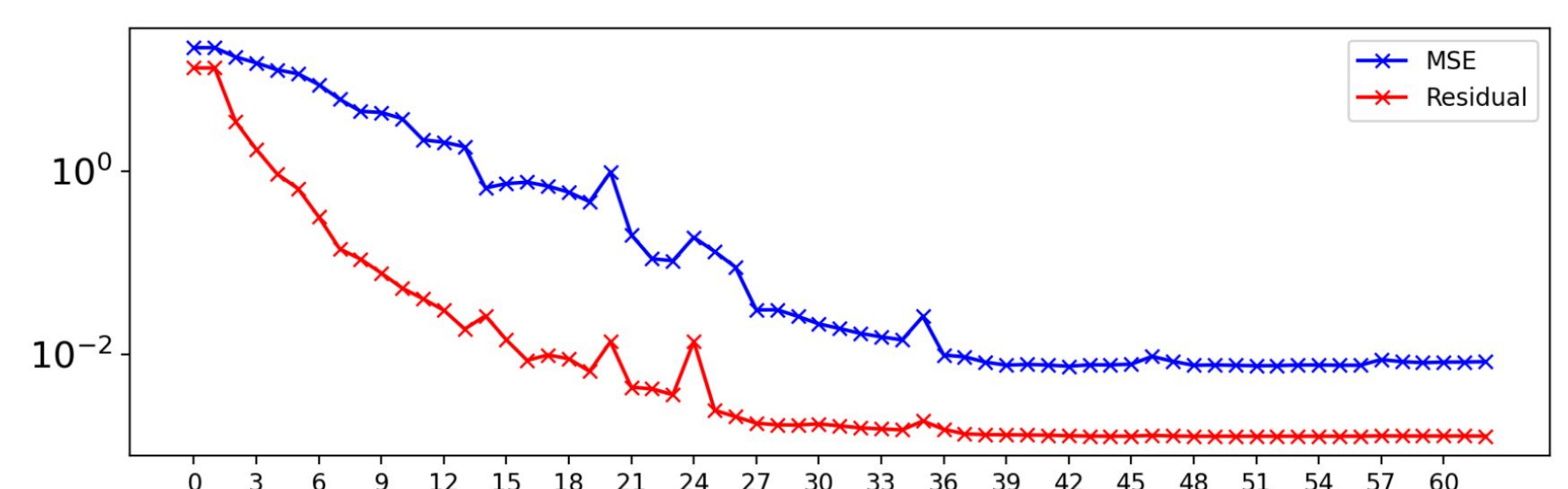
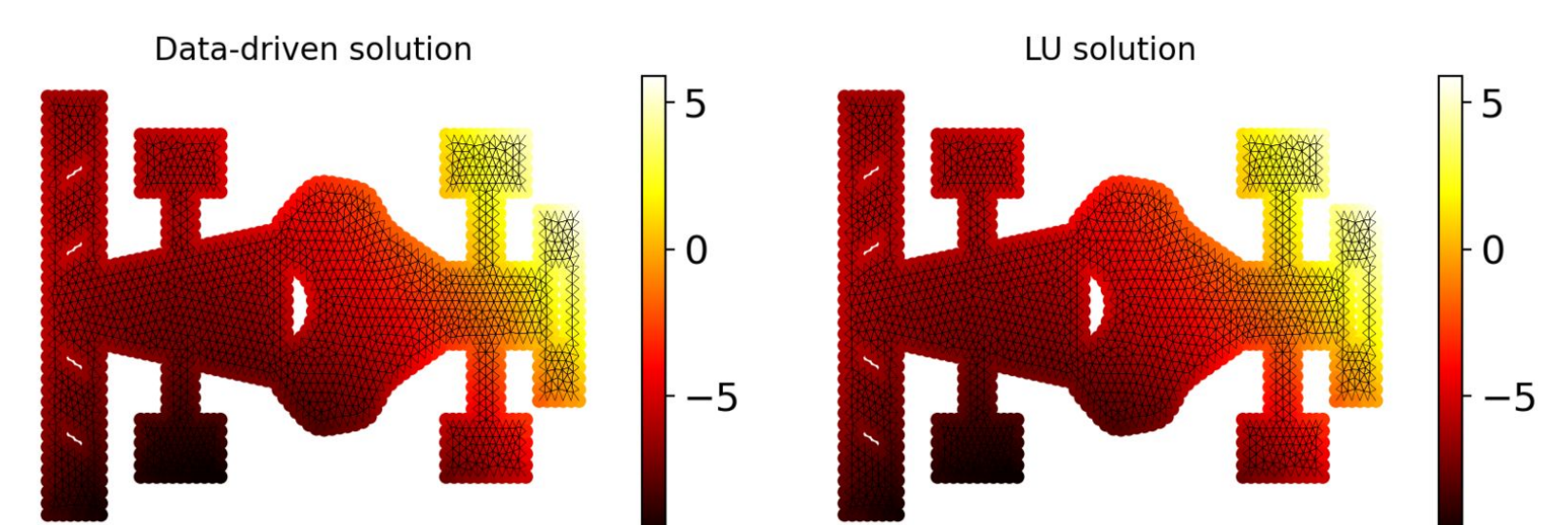
References :

- [1] : G.E.Karniadakis, M.Israeli, S.A.Orszag : High-order splitting methods for the incompressible Navier-Stokes equations.  
[2] : B.Donon, Z.Liu, W.Liu, I.Guyon, A.Marot, M.Schoenauer : Deep Statistical Solver.  
[3] : S.Bai, J.Z.Kolter, V.Koltun : Deep equilibrium model

### Results

Residual	MSE w.r.t LU solution
<b>2.27e-3</b>	<b>1.06e-2</b>

Results obtained from the Test set (unseen geometries and problems)



Nb Nodes	Residual	MSE w.r.t LU	MSE Boundary	MSE Interior
<b>1941</b>	<b>1.28e-3</b>	<b>8.31.e-3</b>	<b>5.9e-7</b>	<b>1.1e-2</b>

Top Left : Data-driven solution to be compared to Top Right : LU solution.  
Bottom : Evolution of the residual and MSE loss across the RootFind solver iterations.

