Machine Learning model for gas-liquid interface reconstruction in CFD numerical simulations with unstructured grid

ML4CFD - DATAIA 2020

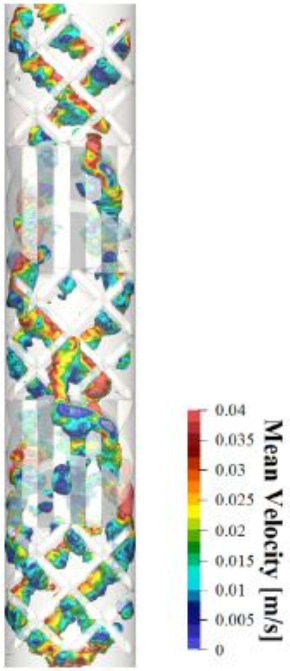
Tamon NAKANO[1], Michele Alessandro BUCCI[1], Jean-Marc GRATIEN[2], Thibault FANEY[2]

[1] INRIA Saclay, Équipe-projet TAU

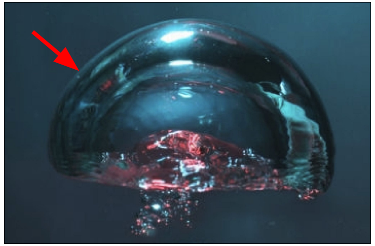
[2] IFPEN, Direction Sciences et Technologie du numérique

Background

- Multiphase flow (ex. gas-liquid phase):
  - pipeline, cooling system, mixer
- Numerical method is an effective approach
- The Volume of Fluid (VoF) method is widely used to model a free-surface
- The VoF reconstructs the surface (i.e., normal, location, curvature) from the volume fraction



Static mixer*



Free surface

*Picture from Marco Scala's PhD thesis at IFPEN Lyon

Navier - Stokes in multiphase flow

Additional terms in the Navier-Stokes equations to deal with multiphase flows:

Pressure Viscosity Surface tension force

Momentum equation

$$\frac{\partial \rho \mathbf{U}}{\partial t} + \nabla \cdot (\rho \mathbf{U}\mathbf{U}) = -\nabla p + \rho \mathbf{g} + \nabla \cdot (\mu \nabla \mathbf{U}) + \rho \mathbf{S} + \sigma \kappa \hat{\mathbf{n}}_S \delta(\mathbf{x} - \mathbf{x}_s)$$

Advection Gravity Extra forces

Continuity equation

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{U}) = 0$$

Density

$$\rho(\mathbf{x}, t) = \begin{cases} \rho_A, & \text{if } \mathbf{x} \text{ in fluid A} \\ \rho_B, & \text{if } \mathbf{x} \text{ in fluid B} \end{cases}$$

Different densities

3

Surface tension force

The surface tension depends on **local geometrical features** of the interface:

Note that:

$$\kappa = \nabla \cdot \hat{\mathbf{n}}_S$$

$$F_S = \sigma \kappa \hat{\mathbf{n}}_S \delta(\mathbf{x} - \mathbf{x}_s)$$

Surface tension coefficient
Constant and depends on the liquid/gas at play

Mean curvature
Local curvature of the interface:

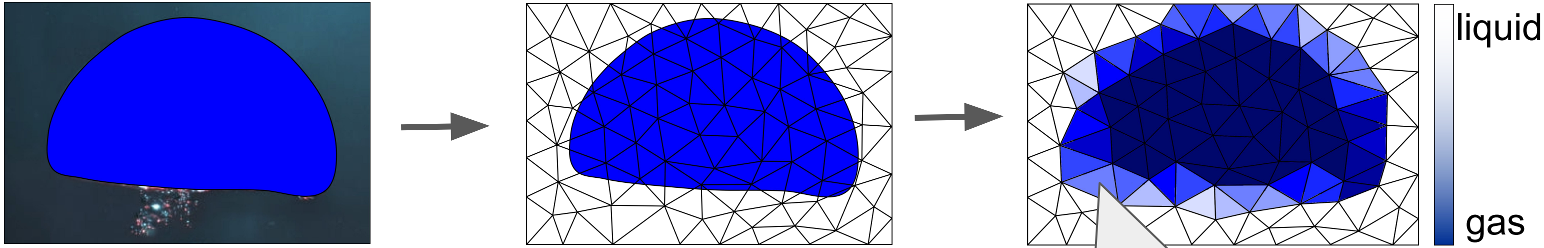
$$\frac{1}{2}\left(\frac{1}{R_1} + \frac{1}{R_2}\right)$$

Normal
Normal direction to the interface in each point of the interface

kronecker delta
The surface tension is active only on the interface

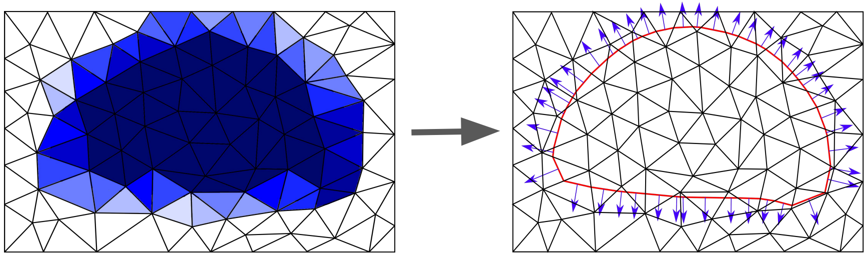
Discretization & Reconstruction

Discretization



Volume fraction, pressure, velocity are stocked

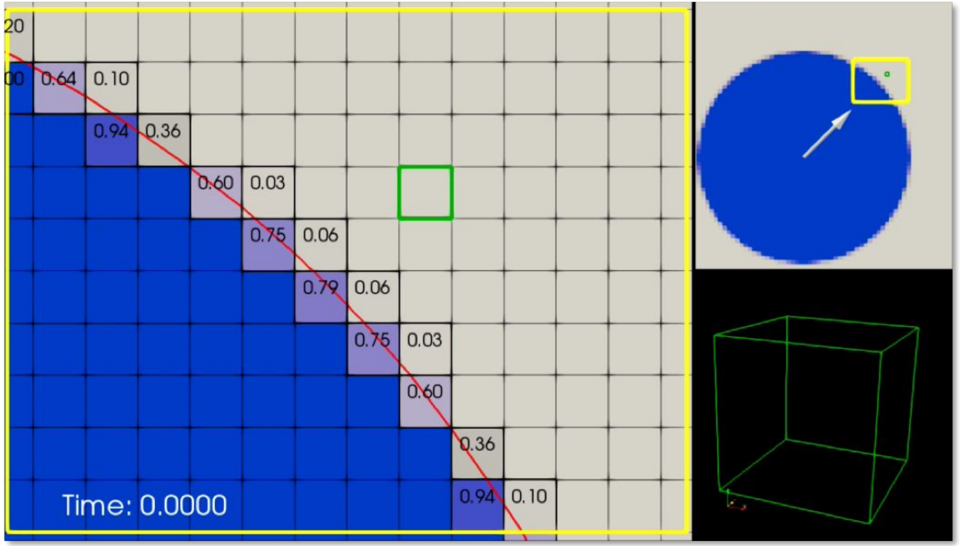
Reconstruction



Geometrical information (normal, curvature etc.) is lost

Several methods of reconstruction (VoF, Level set)

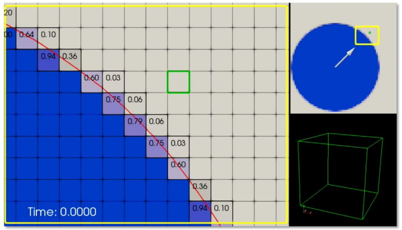
Reconstruction by the VoF



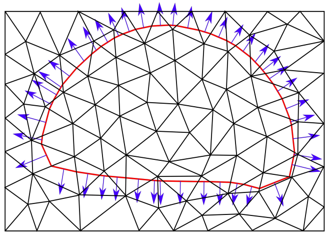
The VoF is **accurate** (good at mass conservation) and performs well on **structured grids**.

State of art and limitations

- The bottlenecks of the VoF
  - High computation cost
  - Bad compatibility with unstructured grids (*e.g.* instabilities and residual currents)
- Neural network approach to reduce the cost
  - Qi et al.(2018): Curvature estimation on 2D surface with Neural Networks (NNs)
  - Patel et al.(2019): Curvature estimation on 3D surface with NNs. Their approach outer-formed the conventional methods



structured grid



unstructured grid

State of art and limitations

- Patel et al.(2019): Multi Layer Perceptron (MLP), emplayable on structured grids with constant discretization.
- Svyetlichnyy et al.(2017): MLP, estimate normal direction on 2D structured grid

	Structured grid	Unstructured grid	Other remarks
VoF	Good	Instability	Expensive
NN	Good	Not available	Less expensive

For industrial cases with complex geometry, unstructured grids are usually employed

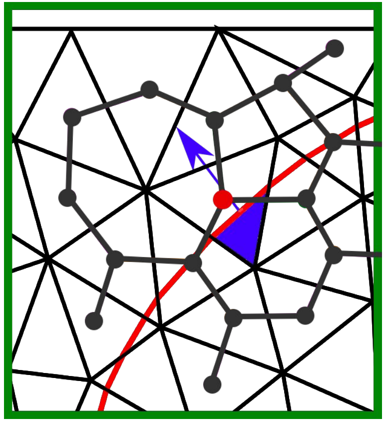
Objective

- We propose a machine learning-based method which is accurate enough to replace conventional reconstruction methods, and is less computational-demanding.

- To that end, we study the use of Graph Neural Network architectures to recover interface properties (e.g., normal vector, curvature, face center, etc.) from the discretized concentration field.

INVENTEURS DU MONDE NUMÉRIQUE

Graph Neural Network (GNN)



GNN can handle a dataset based on graph (*i.e.* nodes and connections between nodes)

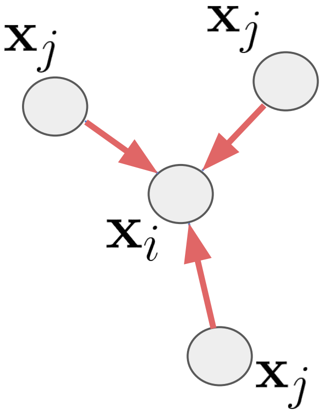
GNNs input:

- Node features: volume fraction α and vertices coordinates for each mesh element
- Edges: element connections

and predict the labels (normal, curvature etc.) on a new (unseen) features

We employ SAGEConv proposed by Hamilton et al.(2017)*

<u>Message passing in SAGEConv</u>



1) $\text{mean}_{j \in \mathcal{N}(i)} \mathbf{x}_j$

2) $\mathbf{W}_2 \cdot \text{mean}_{j \in \mathcal{N}(i)} \mathbf{x}_j$

3) $\mathbf{x}'_i = \mathbf{W}_1 \mathbf{x}_i + \mathbf{W}_2 \cdot \text{mean}_{j \in \mathcal{N}(i)} \mathbf{x}_j$

4) $\sigma \left(\mathbf{x}'_i = \mathbf{W}_1 \mathbf{x}_i + \mathbf{W}_2 \cdot \text{mean}_{j \in \mathcal{N}(i)} \mathbf{x}_j \right)$

Inria INVENTEURS DU MONDE NUMÉRIQUE

*Inductive Representation Learning on Large Graphs
William L. Hamilton, Rex Ying, Jure Leskovec

IFP Energies nouvelles

Synthetic dataset

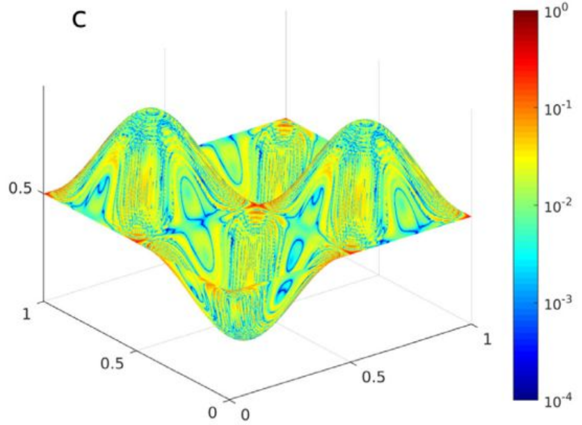
Patel et al. used spherical surfaces

- Larger error at saddle points
- A paraboloid contains this geometry

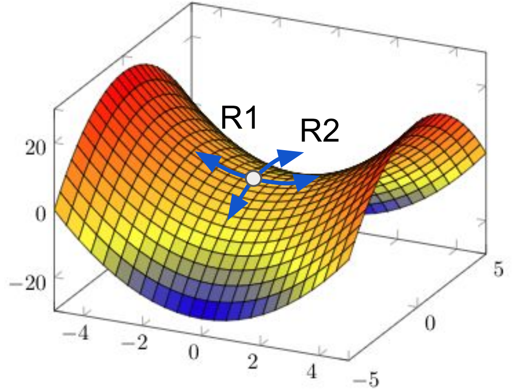
A paraboloid is defined by

$$\frac{x^2}{2R_1} + \frac{y^2}{2R_2} = z$$

R_1, R_2 are the radius on the two parabola at the center which is randomly chosen in space

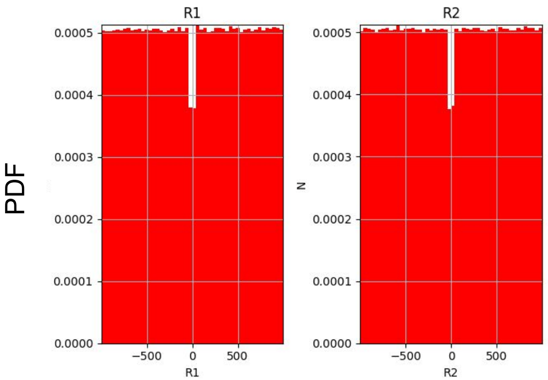


Error on the prediction by ML by Patel et al.(2019)

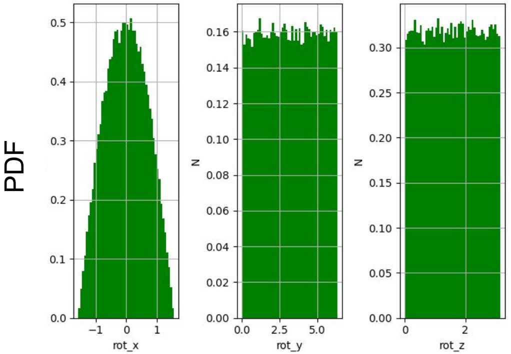


Paraboloid

Distribution in the dataset

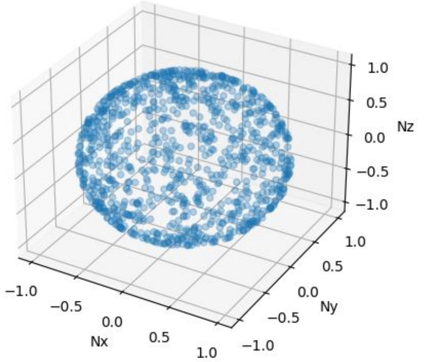


Normalized histogram of R1 and R2

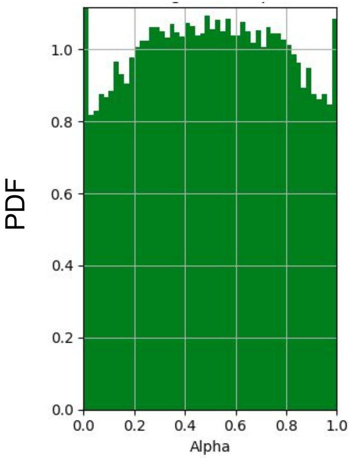


Normalized histogram of $rot_{x,y,z}$

Normal (1k graphs displayed for the simplicity)



Alpha

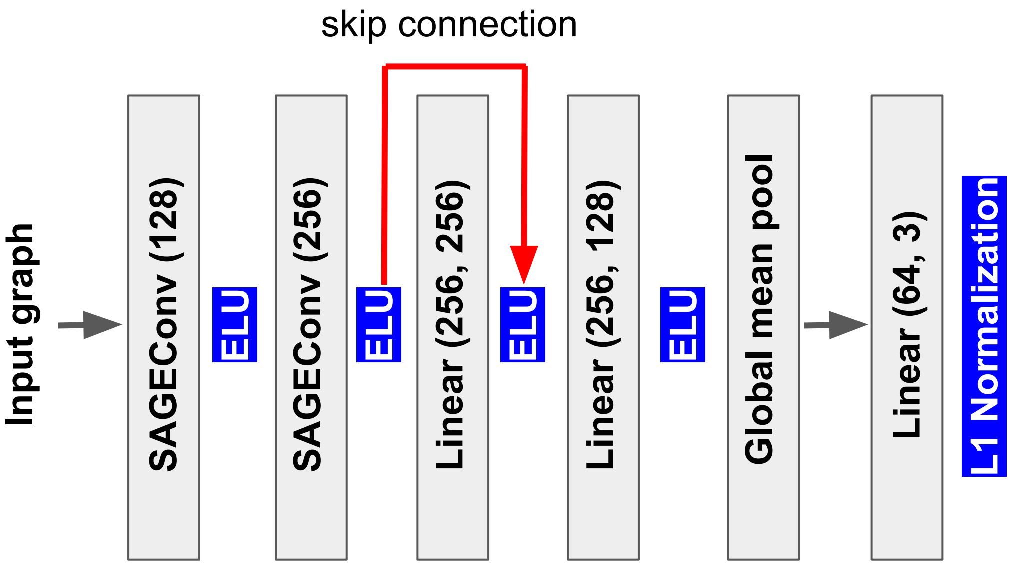


- $R_1, R_2 = \pm[10, 1000]$ (uniforme distribution)
- $(rot_x, rot_y, rot_z) = \cos^{-1}(2u_2 - 1), \pi u_2, 2\pi u_2$ (uniform rotation in 3D)
- Alpha distribution is uncontrollable in advance => quasi-uniform is generated

=> **A non-biased dataset has been then generated**

Architecture

Input
- Coordinate of vertices
- Adjacency matrix
- Alpha



Output:
- Normalize normal vector

Training

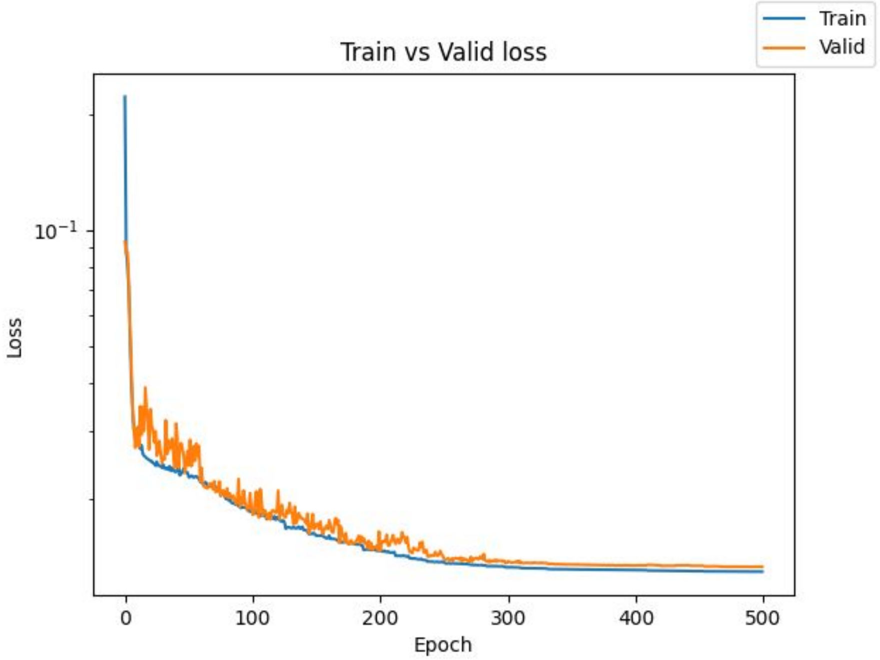
- 100k graphs: each graph has 1 label and corresponding features
- Train/Validation ratio: 8:2
- Variable learning rate (min.1e-6)
- Adam optimiser weight decay: 0.0001
- Loss: L1,

$$Loss = \sum_{i=0}^{n} |y_{true} - y_{pred}| + \lambda \sum_{i=0}^{n} w_i$$

$$\lambda = 0.0001$$

The loss at 500th epoch

- Training: 1.30e-02
- Validation: 1.34e-02



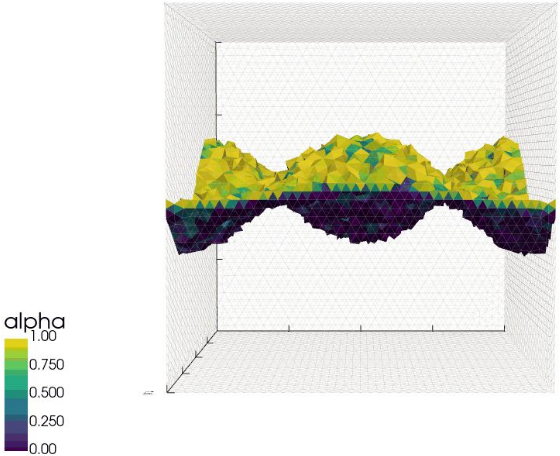
Generalization

Predict the normal on a **new surface** different from the paraboloid for the training:

$$f(x, y) = 0.3\sin(3x)\cos(3y)$$

$(R/\Delta x)_{max} = 8.33$ (a bit smaller than the covering range)

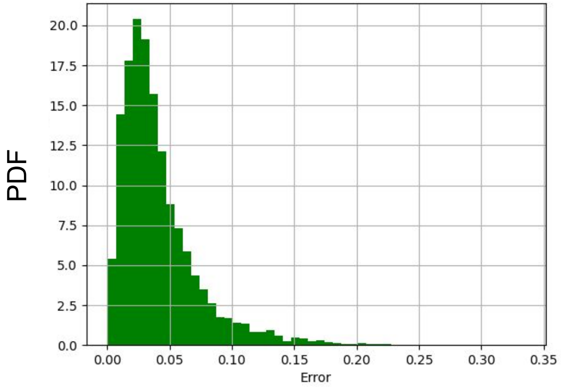
$$(R/\Delta x)_{training} = \pm[10, 1000]$$



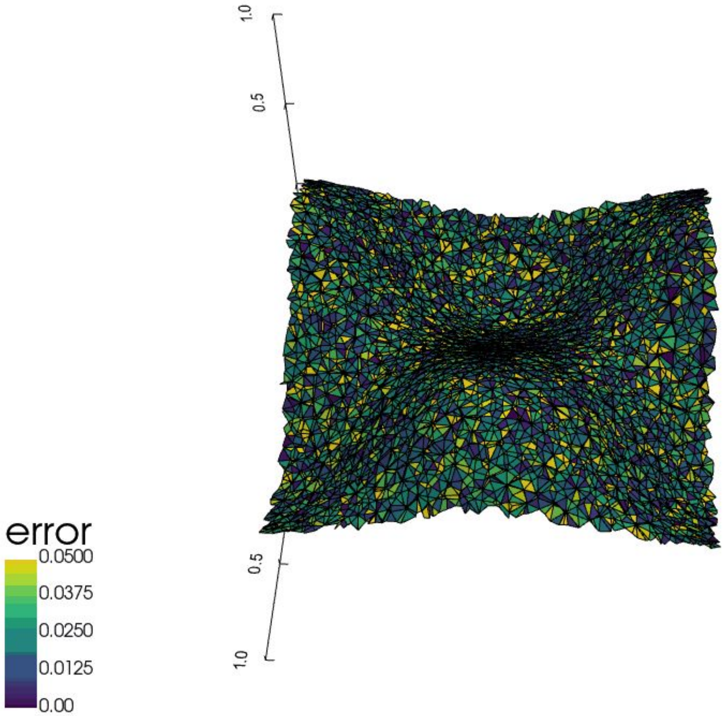
Generalization

Error on $\|N_{label} - N_{pred}\| \approx N_{label} \cdot N_{pred}$

- Average error: 4.14e-2
- Uniform, no dependence on the geometry
- Smooth distribution, no outlier



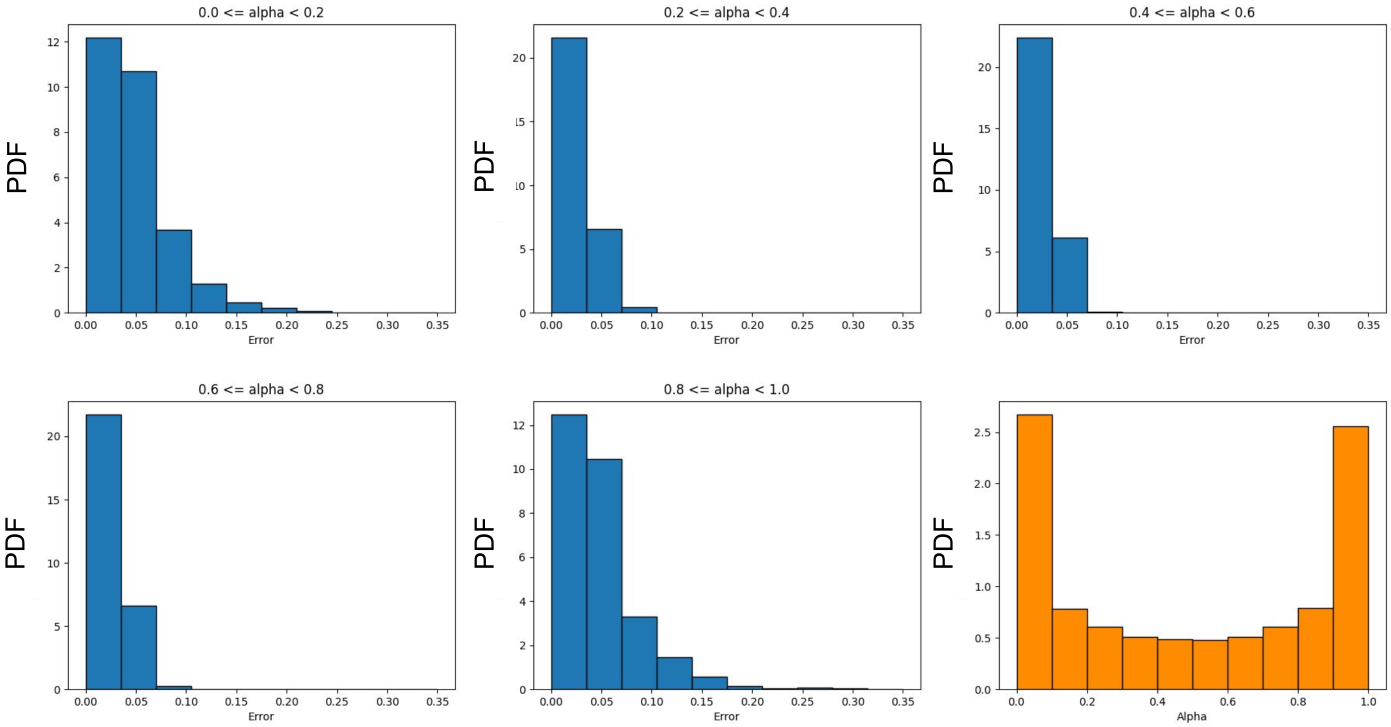
Normalized histogram of the error



Predicted surface

Analysis on the error

Normalized histogram on the error on different ranges of Alpha



Normalized histogram of Alpha on the test surface

- Larger error at the extreme values of Alpha
- More samples at the extreme Alpha in the sinusoidal surface case
- Possible approach: Design an ad-hoc architecture to improve the prediction on the extreme Alpha values

Conclusions & Perspectives

Conclusion

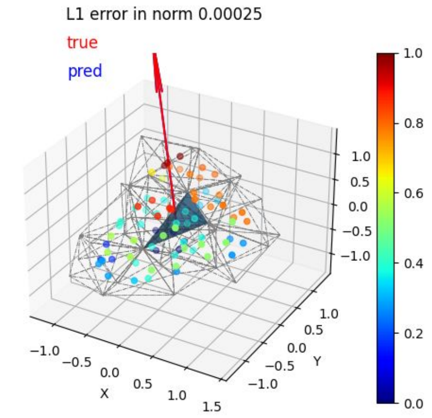
1. **Graph Neural Network** can be an alternative to the conventional surface reconstruction methods in terms of accuracy
2. It is more stable than the standard VoF

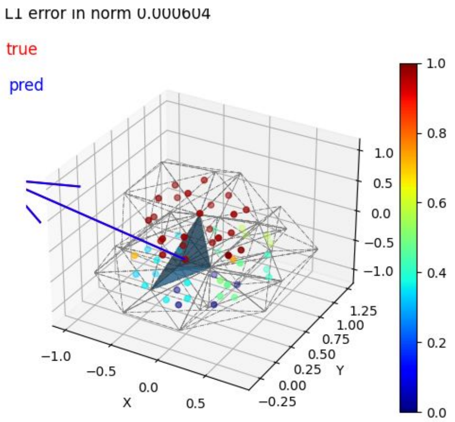
Perspectives

1. Increase the amount of data to improve the accuracy of the model
2. Improve the prediction at the extreme value of Alpha
3. Prediction on the curvature and the center
4. Treatment of boundary conditions
5. Employ the model in a CFD code (i.e. OpenFOAM)
6. Assessment of time performance of the model

Analysis on error

Smallest error





Largest error