NEURAL INFORMATION PROCESSING SYSTEMS

Paper

Webpage

Towards Universal Mesh Movement Networks

NeurIPS 2024 (Spotlight)

Mingrui Zhang[1], Chunyang Wang[1], Stephan C. Kramer[1], Joseph G. Wallwork[2], Siyi Li[1], Jiancheng Liu[3], Xiang Chen[4], Matthew D. Piggott[1]

[1]Imperial College London, [2]University of Cambridge, [3]Michigan State University, [4]Huawei, Noah's Ark Lab

IMPERIAL UNIVERSITY OF CAMBRIDGE MICHIGAN STATE UNIVERSITY HUAWEI

Numerical Simulation

Numerical Physical Simulation

Physical knowledge (e.g., Governing Equation) -> Solutions

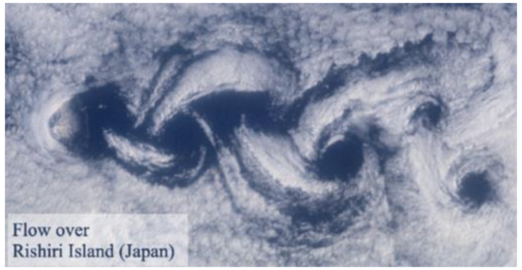
Pros: Physical plausible, generalizable
Cons: Computational expensive, discretization errors, undiscovered physics

Governing Equation → Discretization → PDE Solver → Solutions

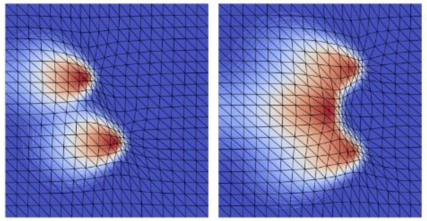
Insufficient knowledge to model the phenomenon (e.g., turbulence modelling still relies on empirical models)

Imperfect discretization schemes (both spatially and temporally) induce errors

Solving large scale especially non-linear system is slow to converge



photograph from NASA, 2001; STS-100



Numerical Simulation

Numerical Physical Simulation

Physical knowledge (e.g., Governing Equation) -> Solutions

Pros: Physical plausible, generalizable

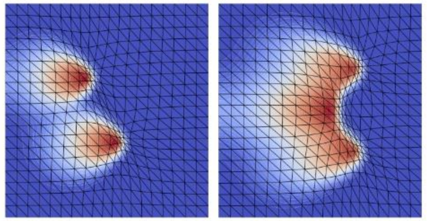
Cons: Computational expensive, discretization errors, undiscovered physics

Governing Equation → Discretization → PDE Solver → Solutions

Insufficient knowledge to model the phenomenon (e.g., turbulence modelling still relies on empirical models)

photograph from NASA, 2001; STS-100

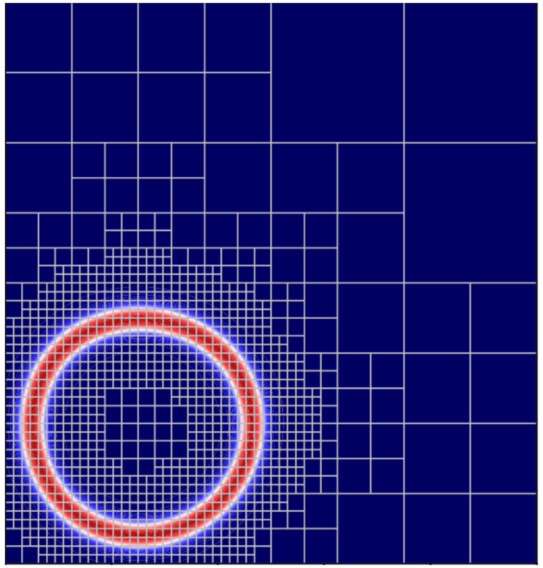
Imperfect discretization schemes (both spatially and temporally) induce errors



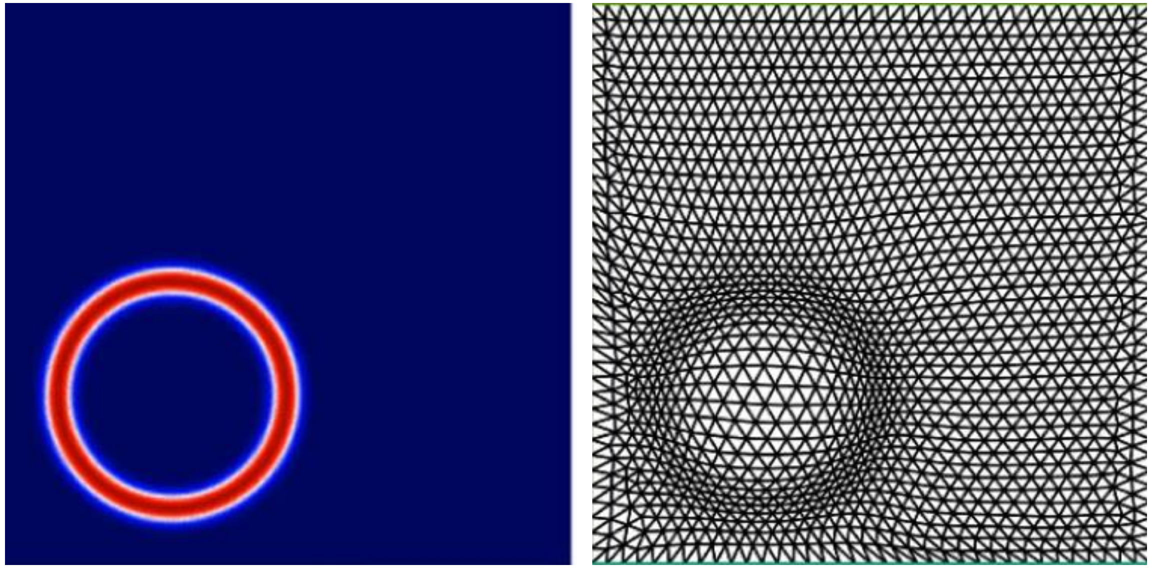
Solving large scale especially non-linear system is slow to converge

Mesh Adaptation

- Mesh adaptation are techniques for distributing mesh according to the requirements of numerical accuracy. Two main categories of mesh adaptation techniques can be identified: h-adaptation and r-adaptation.



h-adaptation



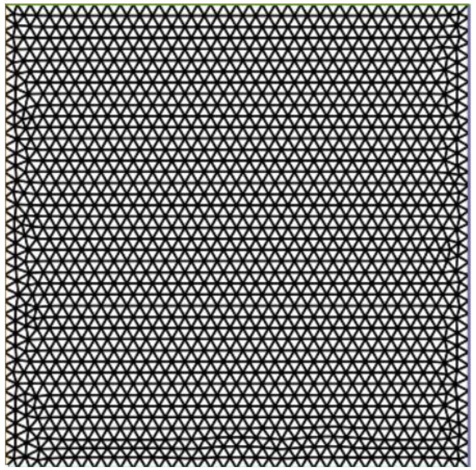
r-adaptation

Corbin Foucart, Aaron Charous, and Pierre F.J. Lermusiaux. Deep reinforcement learning for adaptive mesh refinement. Journal of Computational Physics, 491:112381, 2023.

Mesh Movement

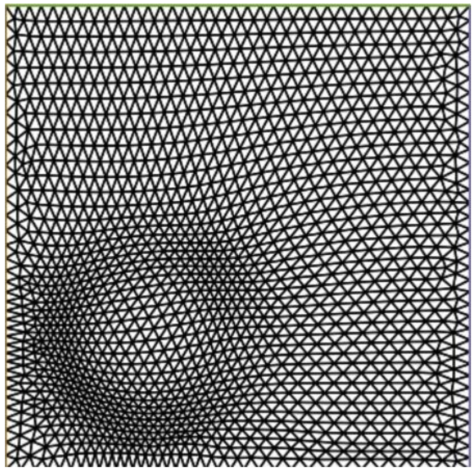
- A monitor function m over the spatial domain is used to specify the desired mesh density. Finding a mapping so that m is equidistributed over the adapted (i.e., physical) mesh.

$$m(\boldsymbol{x})\det(\boldsymbol{J}) = const. \qquad \boldsymbol{J} = \frac{\partial f(\boldsymbol{\xi})}{\partial \boldsymbol{x}}$$



Ω_c

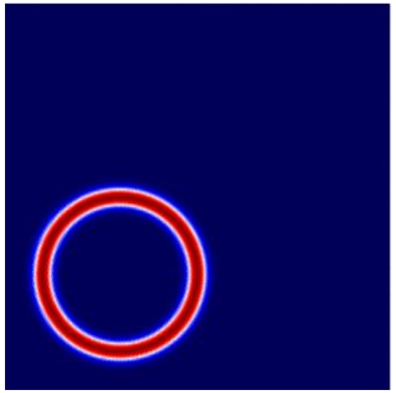
Goal $\boldsymbol{x} = f(\boldsymbol{\xi})$



Ω_p

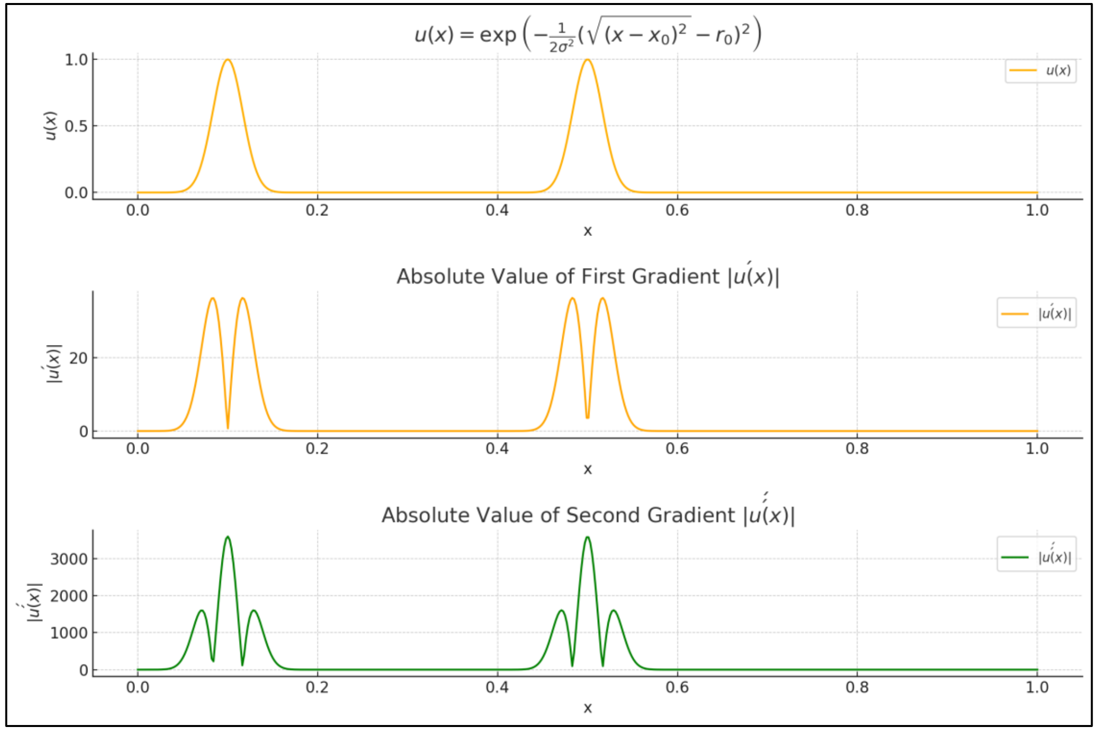
Monitors

- A monitor function m over the spatial domain is used to specify the desired mesh density. Finding a mapping so that m is equidistributed over the adapted (i.e., physical) mesh.



Solution

Monitor Function
e.g., gradient, hessian etc

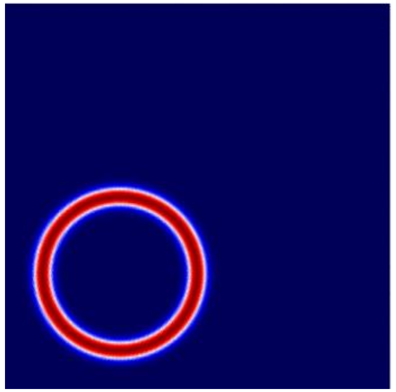


Monitors

Monitors

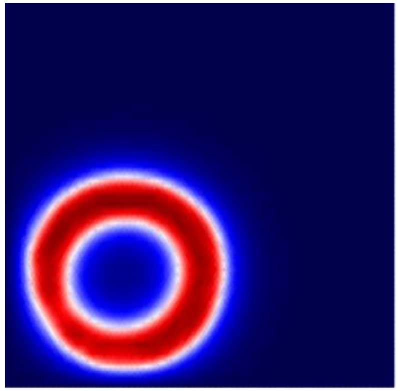
- A monitor function m over the spatial domain is used to specify the desired mesh density. Finding a mapping so that m is equidistributed over the adapted (i.e., physical) mesh.

$$m(\mathbf{x}) = 1 + \max\left(\alpha \frac{\|H(u)\|}{\max \|H(u)\|}, \beta \frac{\|G(u)\|}{\max \|G(u)\|}\right)$$

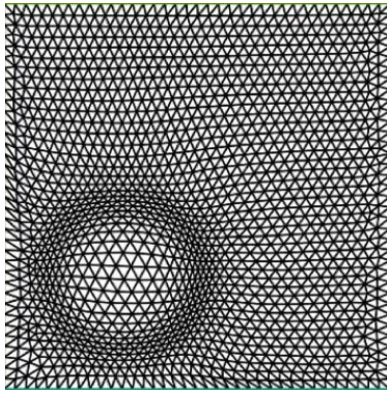


Solution

e.g., Combined gradient and hessian with smoothing →



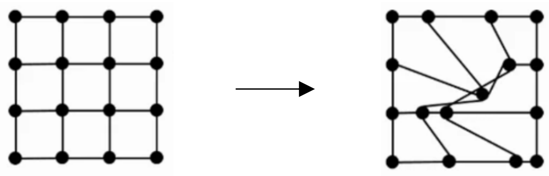
Monitors



Adapted mesh

Mesh tangling

- Mesh tangling occurs when elements of a computational mesh overlap or intersect, i.e., negative Jacobians or inverted elements.



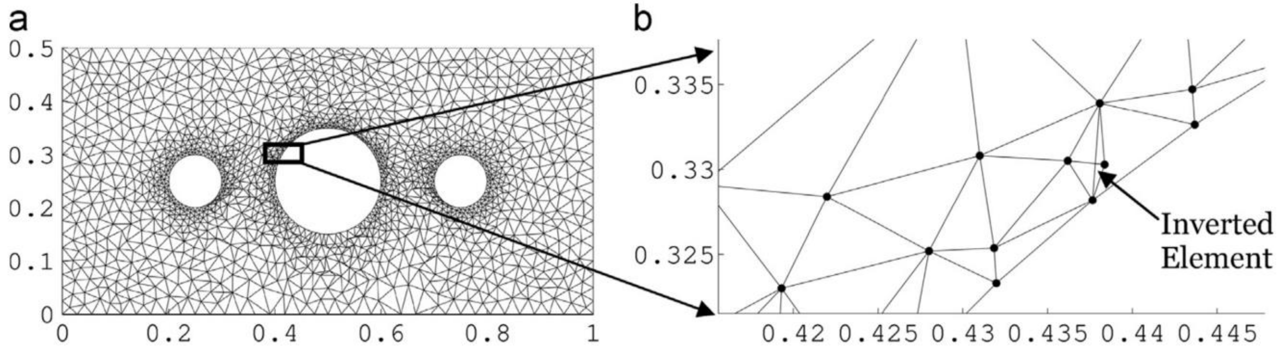
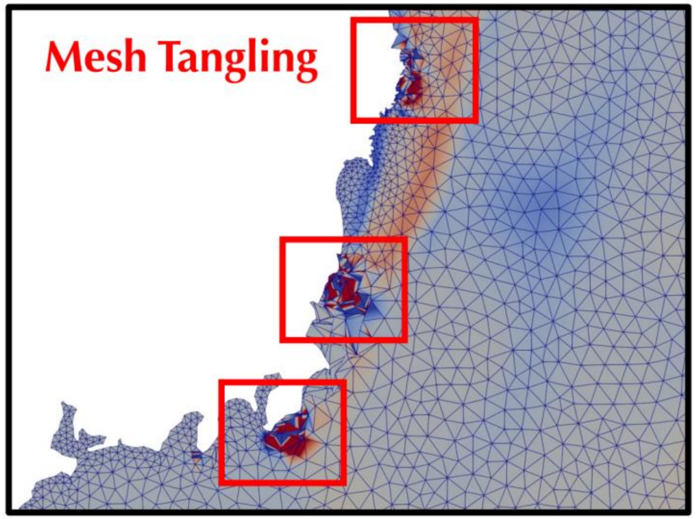


Fig. 1. (a) A tangled mesh and (b) a close-up of the tangling (observe the inverted element).



Mesh tangling leads divergence in simulations!

Josh Danczyk, Krishnan Suresh,Finite element analysis over tangled simplicial meshes: Theory and implementation, Finite Elements in Analysis and Design, Volumes 70–71,2013

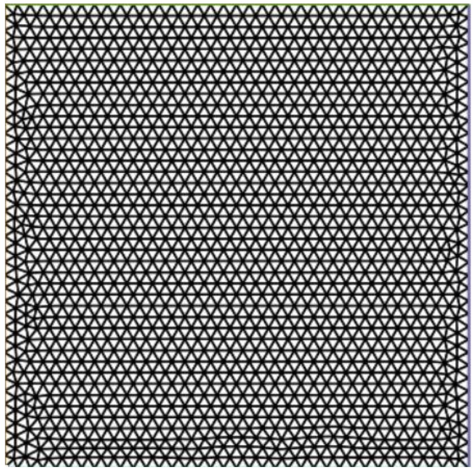
Monge-Ampère Mesh Movement

- By using concepts from **optimal transport theory** , the problem can be constrained to have a unique solution, with the deformation of the map expressed as the gradient of a scalar potential ϕ, i.e., $\boldsymbol{x}(\boldsymbol{\xi}) = \boldsymbol{\xi} + \nabla_{\xi}\phi(\boldsymbol{\xi})$:

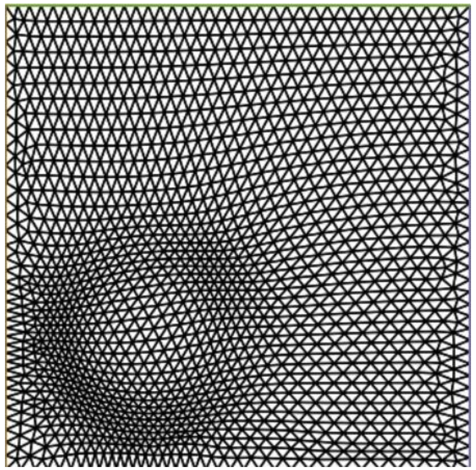
$$m(\boldsymbol{x})\det(\boldsymbol{J}) = const. \qquad \boldsymbol{J} = \frac{\partial f(\boldsymbol{\xi})}{\partial \boldsymbol{x}}$$

$$m(\boldsymbol{x})\det(\boldsymbol{I} + \boldsymbol{H}(\phi(\boldsymbol{\xi}))) = const. \quad \boldsymbol{H}(\phi)_{ij} = \frac{\partial^2 \phi}{\partial \xi_i \partial \xi_j}$$

Goal $\boldsymbol{x} = f(\boldsymbol{\xi})$



Ω_c

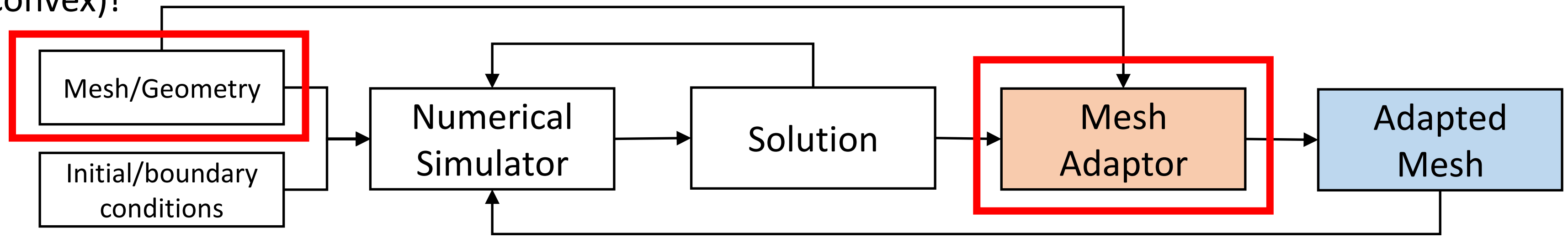


Ω_p

Monge-Ampère Mesh Movement

- **Pros**
  - Output a non-tangled adapted mesh which satisfies equal-distributed theory
- **Cons**
  - Computational expensive: we need to solve an additional non-linear PDE!
  - Complex geometry: Monge-Ampère fails on scenarios with complex geometries.

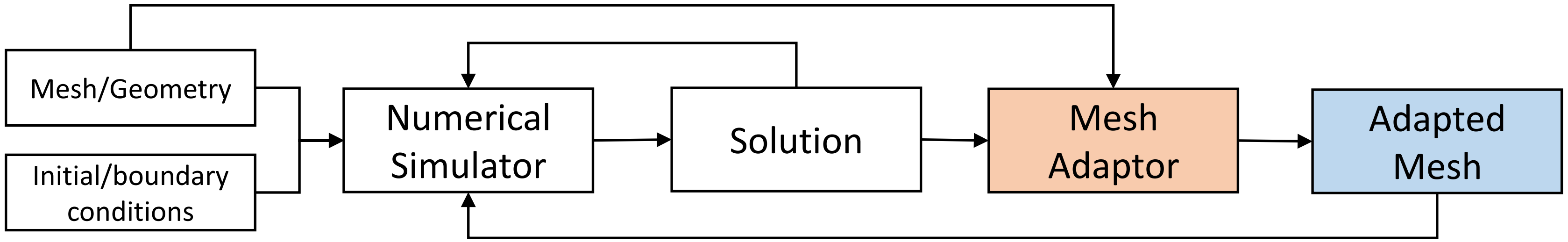
Geometry can not be too complex
(i.e., non-convex)!



MA PDE solver is too expensive!

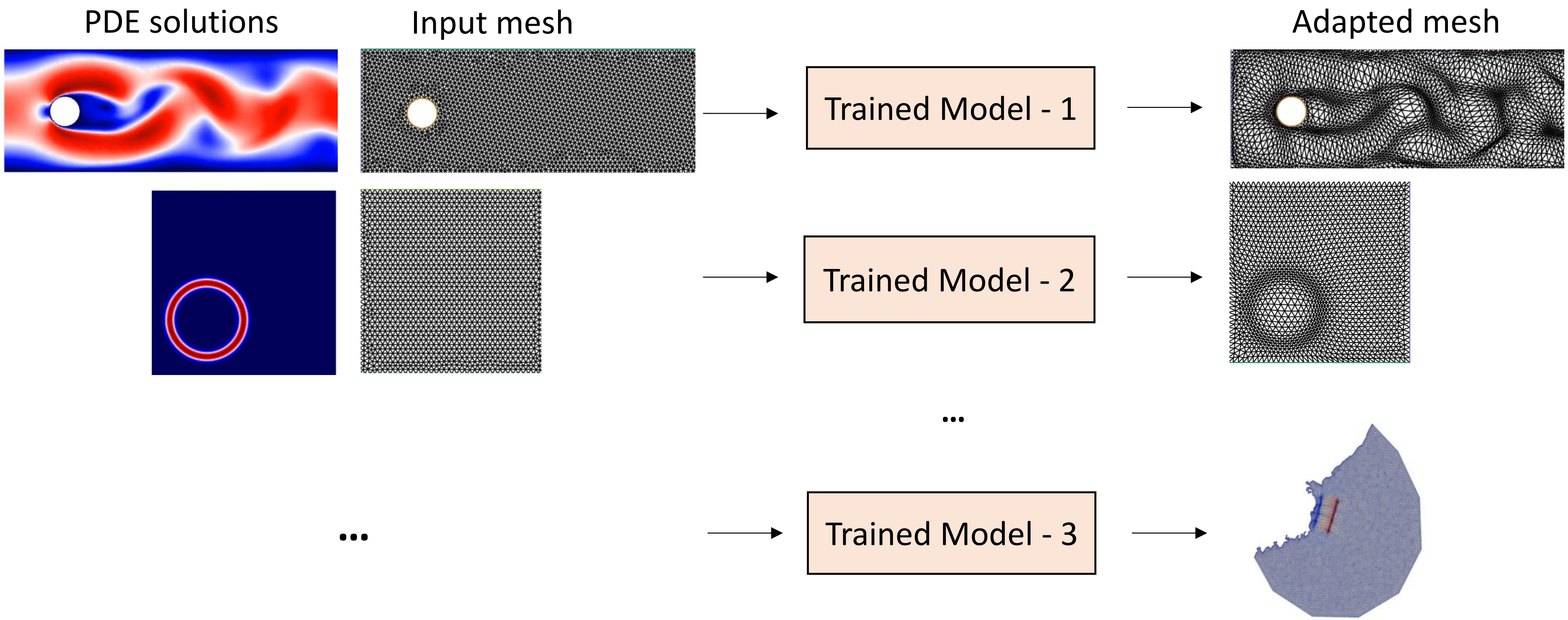
Learning based mesh adaptation

- Goals:
    - **Mesh adaptor** is efficient
    - **Adapted mesh** is non-tangled
    - **Geometry** can be complex (e.g., non-convex)



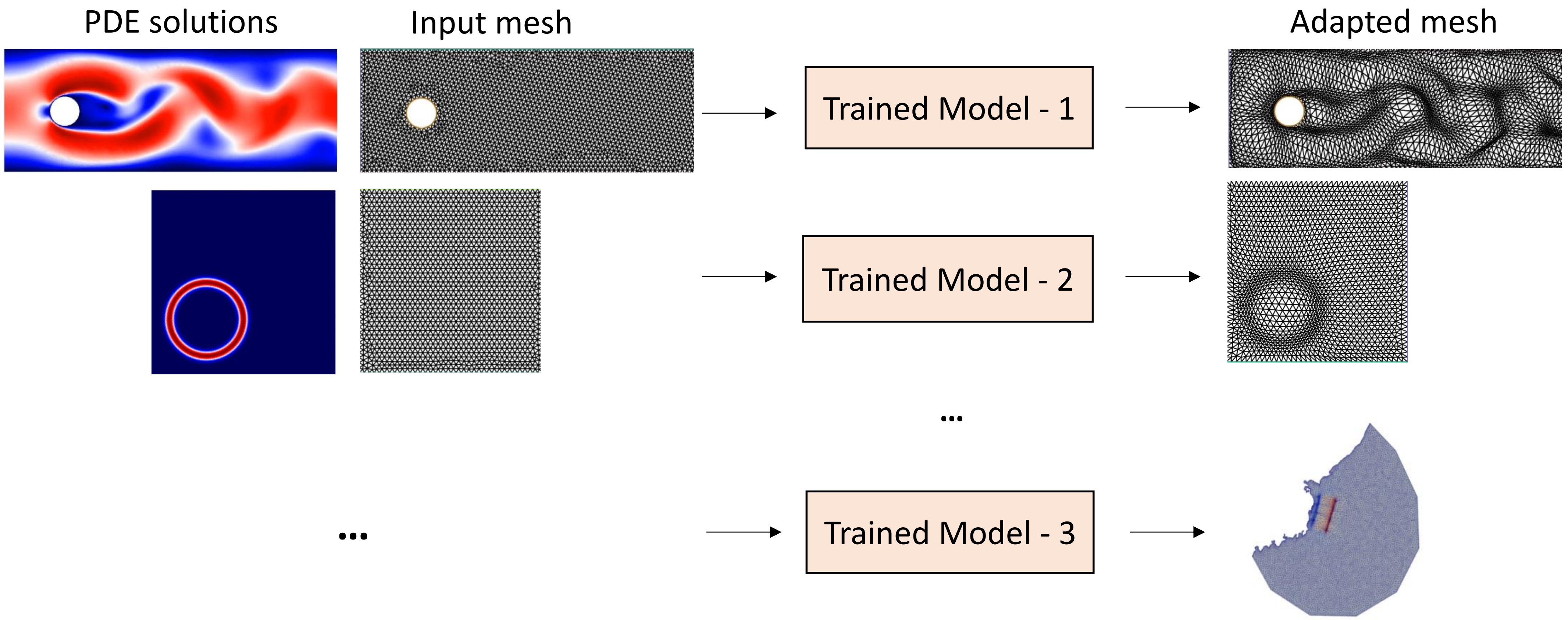
Limitation of Existing Work

- Require to re-train the model from scratch given a new problem or scenario



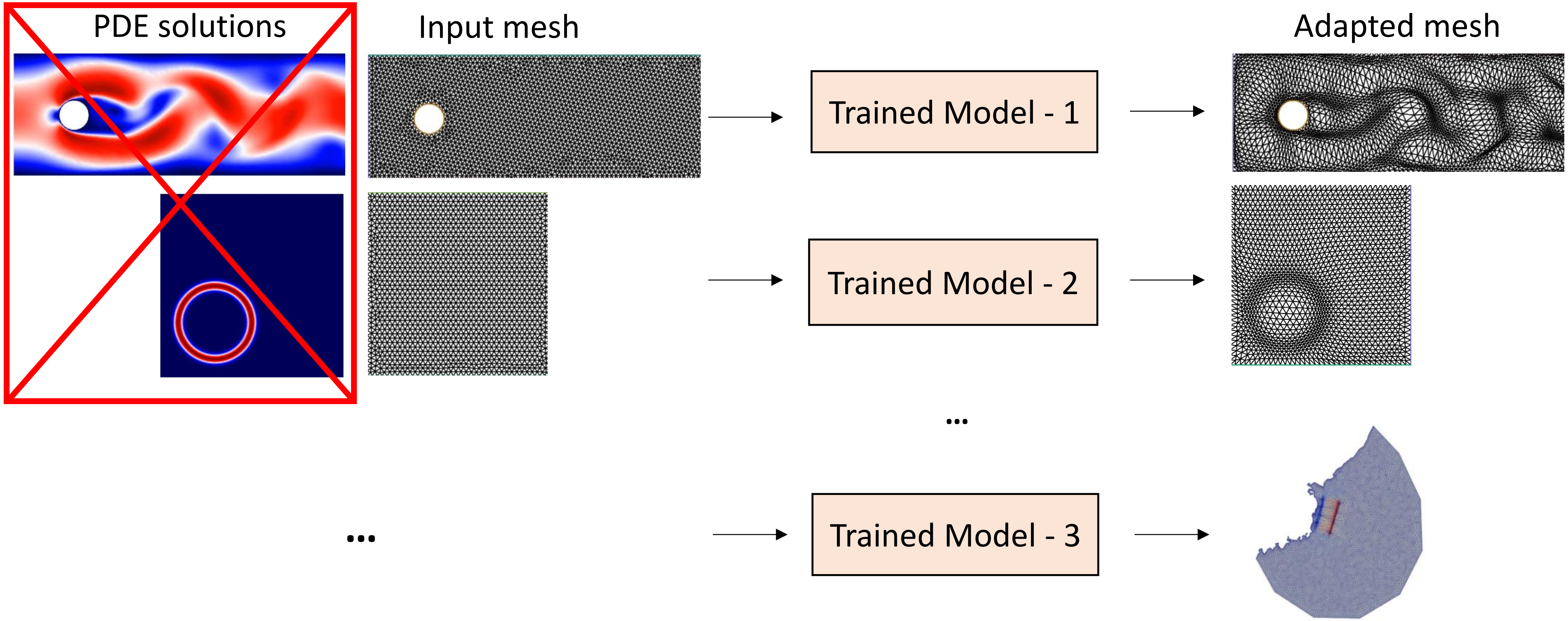
Limitation of Existing Work

- Can we train once and and zero-shot apply the trained model to other problems or scenarios?



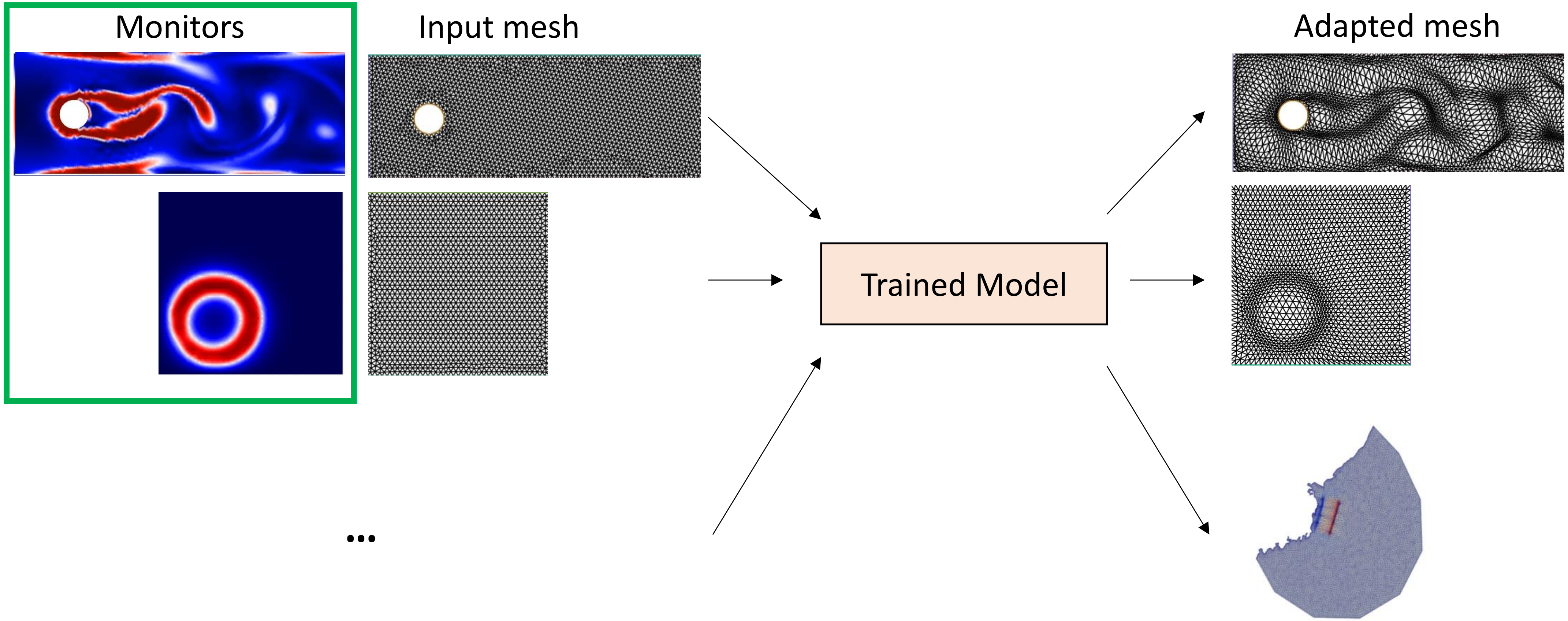
Key Intuition

- **Decouple the inputs from the PDEs, use monitors instead of PDE solutions**



Key Intuition

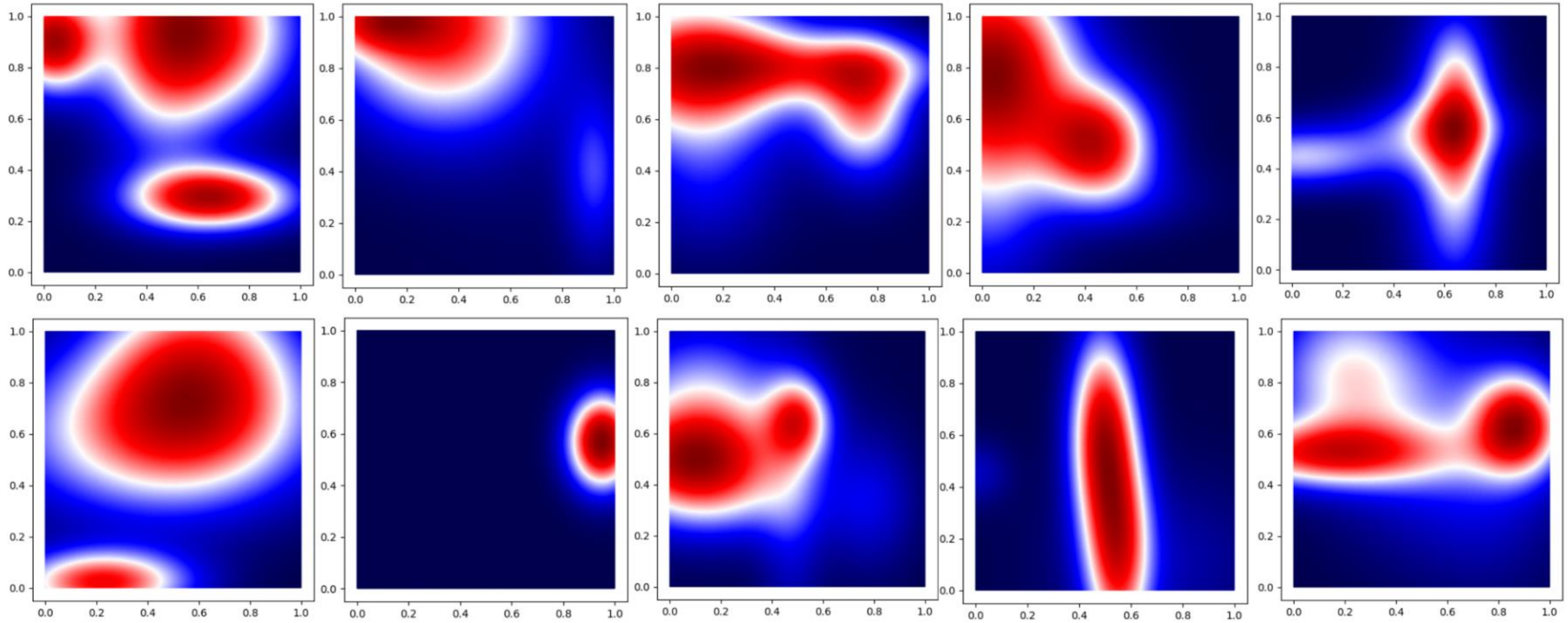
- **Decouple the inputs from the PDEs, use monitors instead of PDE solutions**



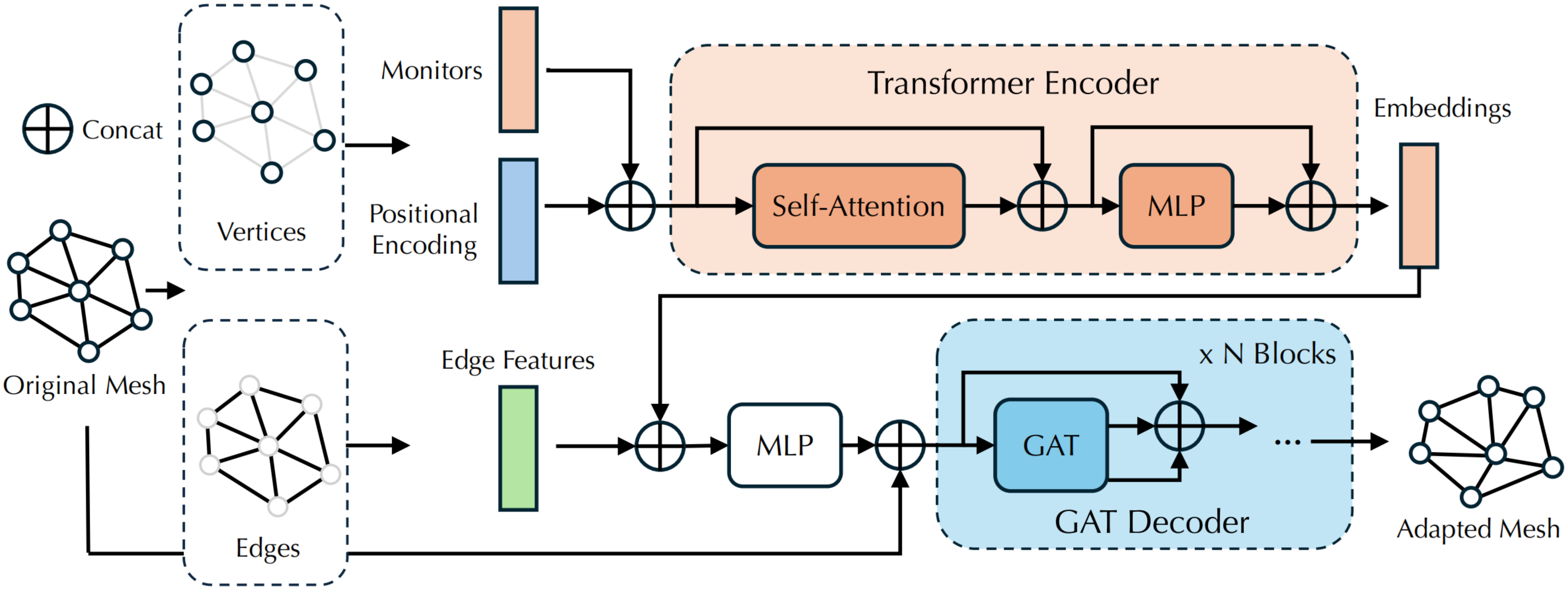
PDE-independent training

- Randomly generating generic fields as monitors for PDE-independent training

$$u = \sum_{k=1}^{N} \exp\left(\frac{(x-\mu_x)^2}{{\sigma_x}^2} + \frac{(y-\mu_y)^2}{{\sigma_y}^2}\right)$$



Networks


Concat
Vertices
Monitors
Positional
Encoding
Transformer Encoder
Self-Attention
MLP
Embeddings
Original Mesh
Edges
Edge Features
MLP
x N Blocks
GAT
...
GAT Decoder
Adapted Mesh

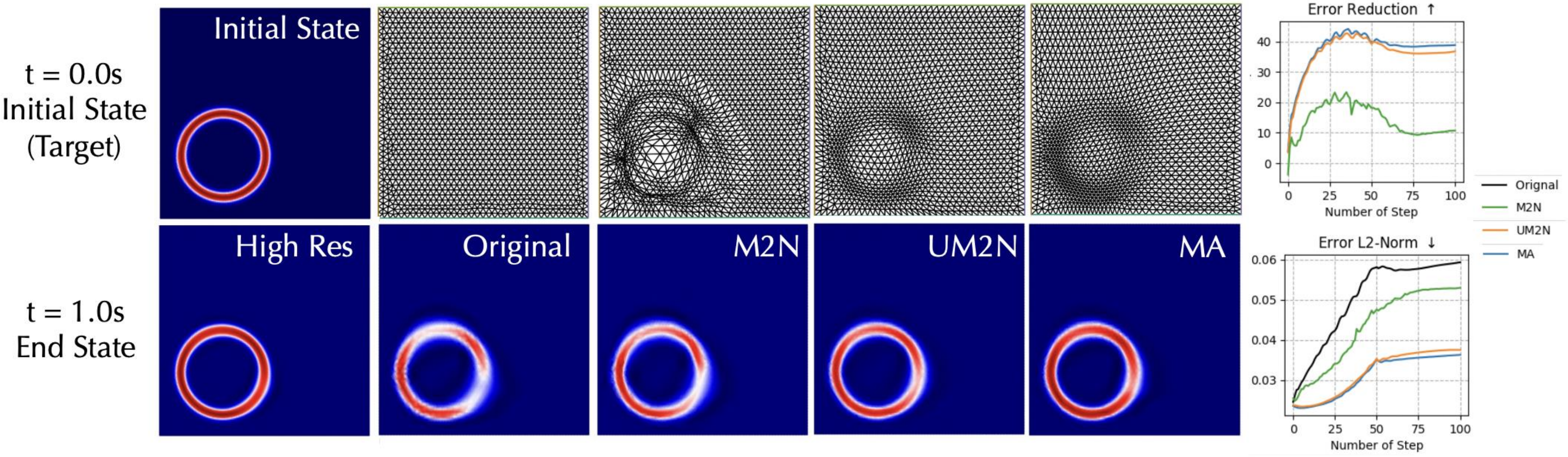
Results on Swirl

Initial condition:

$$u_0 = \exp\left(-\frac{1}{2\sigma^2}\left(\sqrt{(x-x_0)^2+(y-y_0)^2}-r_0\right)^2\right)$$

Velocity field:

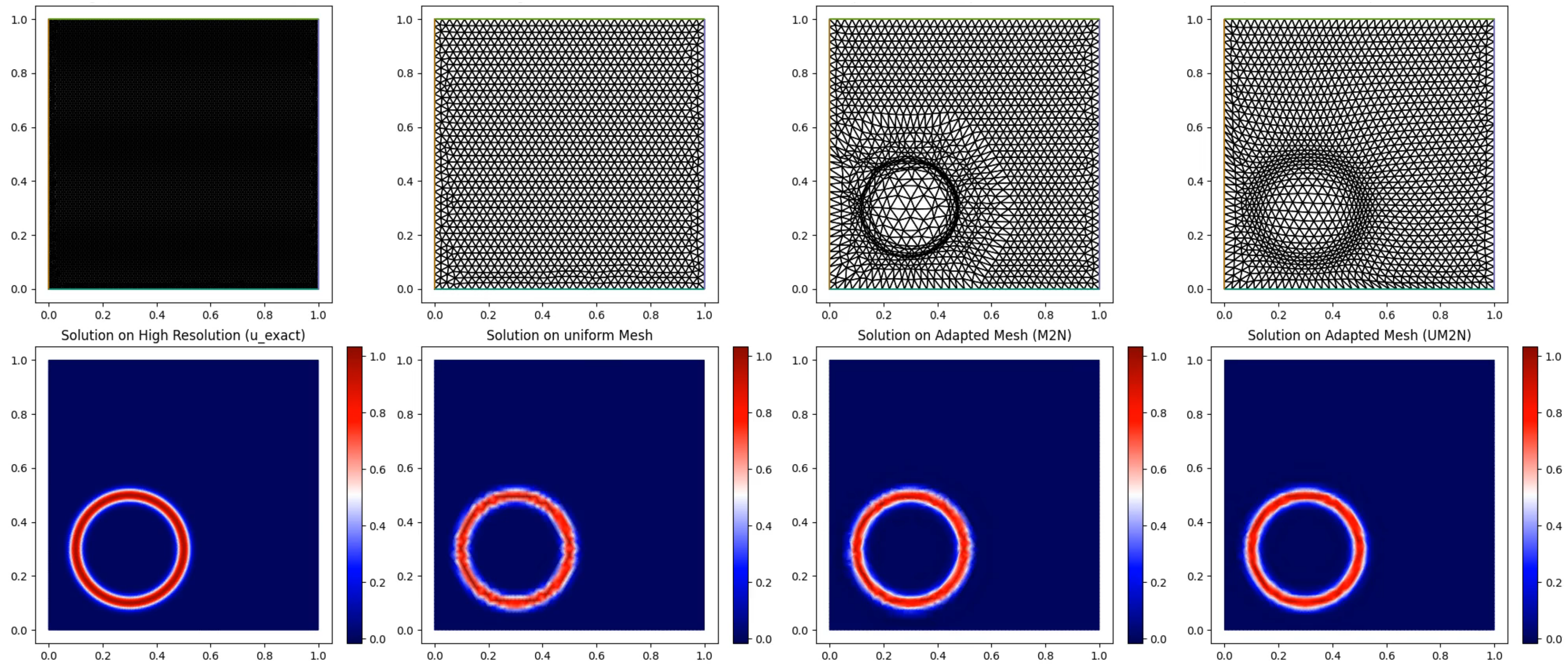
$$\boldsymbol{c}(x,y,t) = \left(\frac{9}{10}a(t)\sin^2(\pi x)\sin(2\pi y), -\frac{9}{10}a(t)\sin^2(\pi y)\sin(2\pi x)\right)$$



Results on Swirl

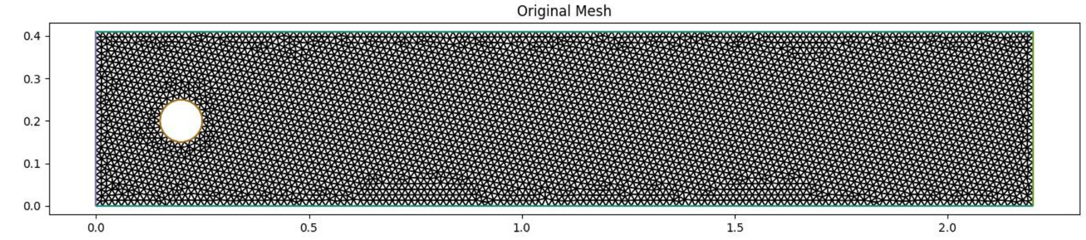
Adapted Mesh (M2N)

Adapted Mesh (UM2N)

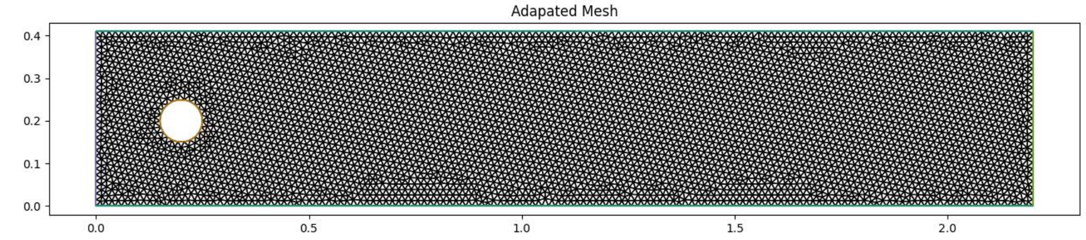


Flow Past a Cylinder

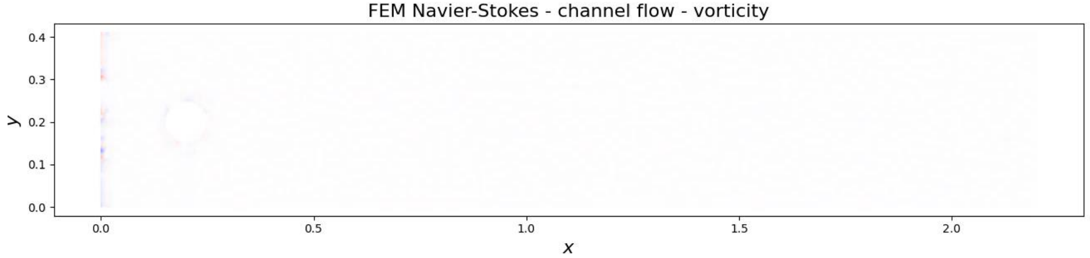
Original Mesh



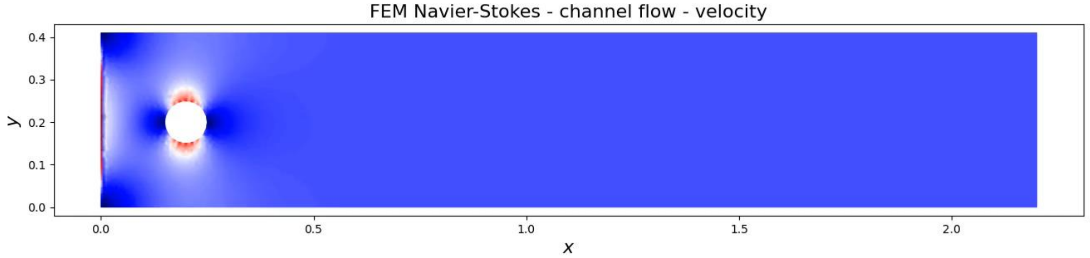
Adapted Mesh (UM2N)



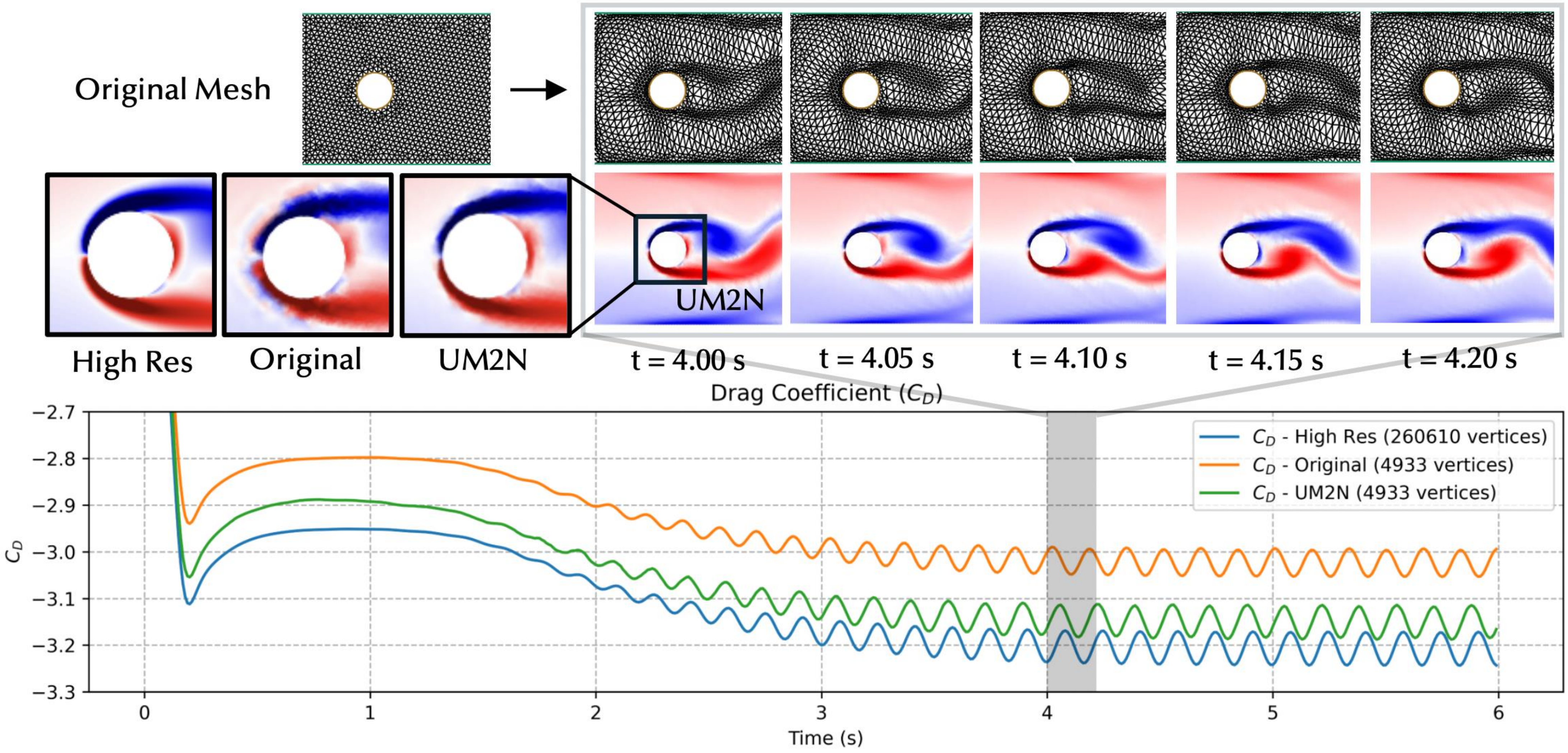
Vorticity



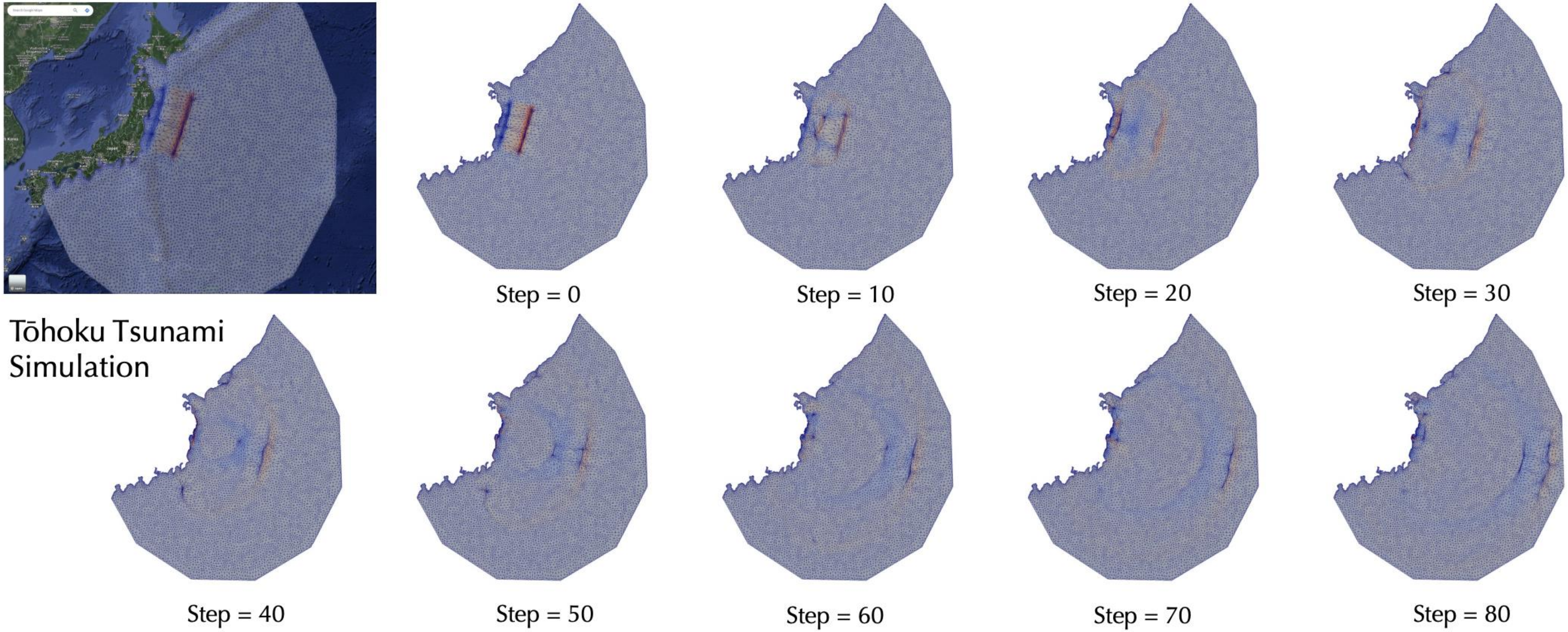
Velocity



Results on Flow Past Cylinder

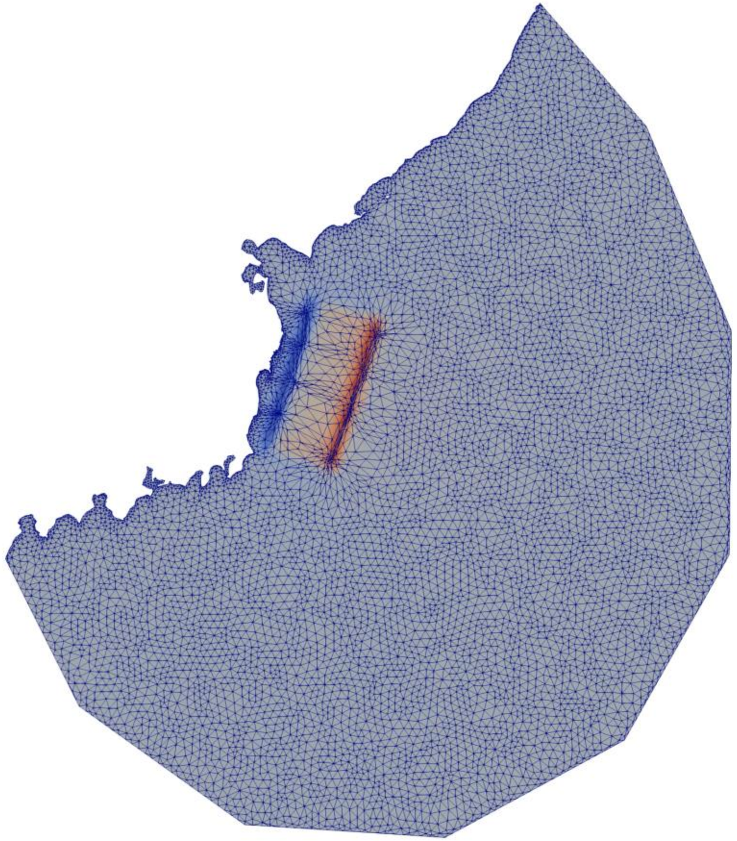


Results on Tsunami Simulation

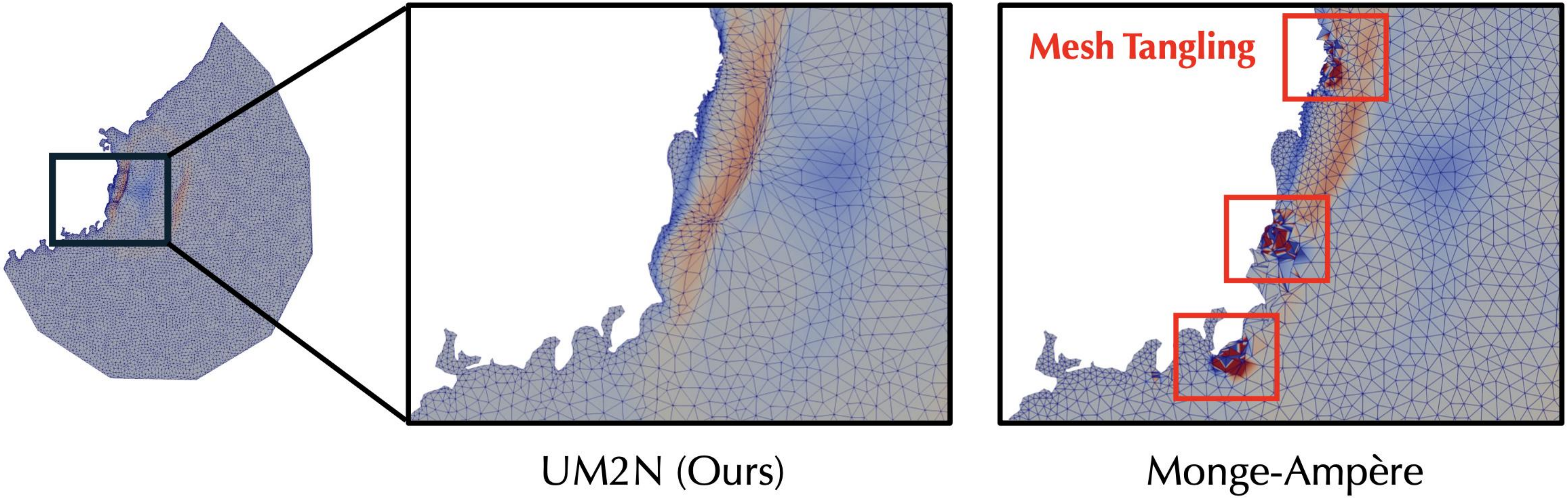


Tohoku Tsunami Simulation

Adapted Mesh
(UM2N)



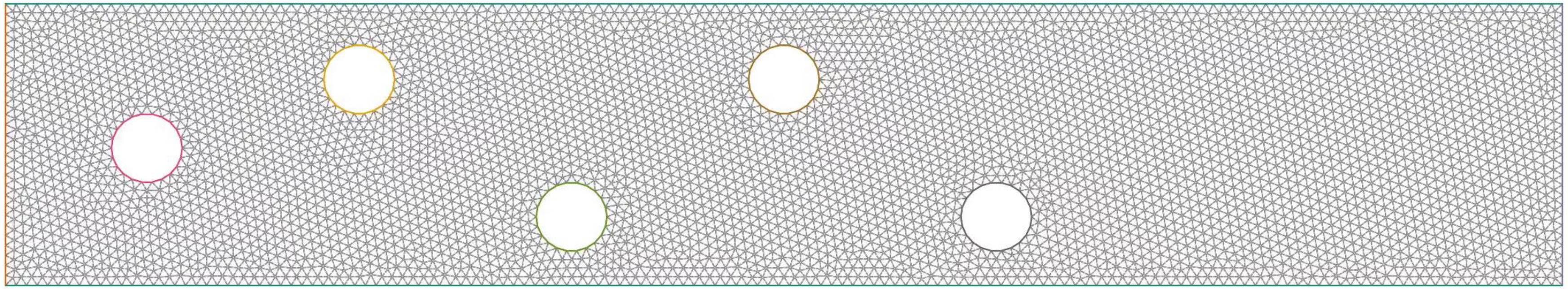
Results on Tsunami Simulation



More Results

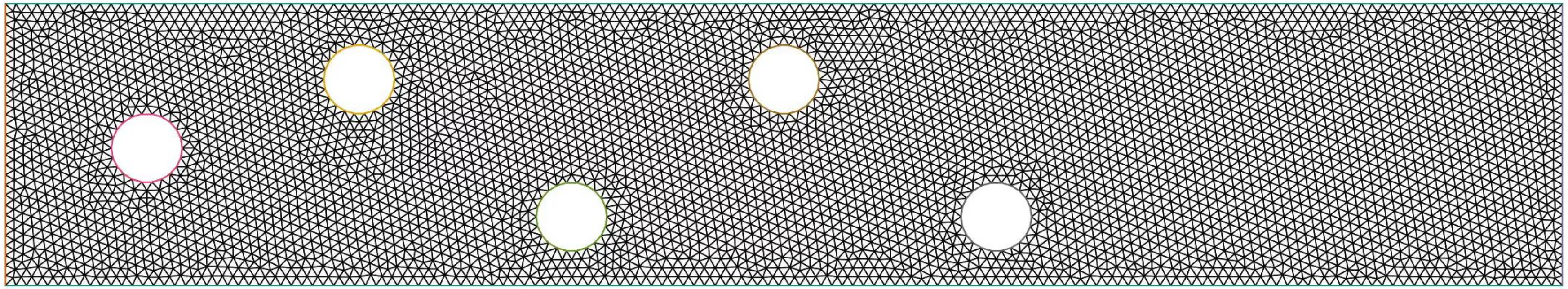
Navier-Stokes - Vortex

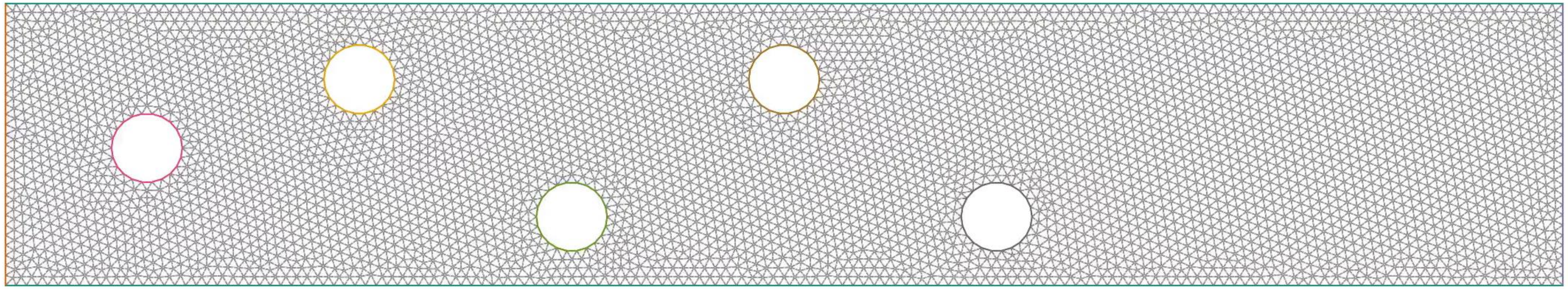
Vorticity



Adapated Mesh (UM2N Output)

Adapted Mesh (UM2N)





Thanks!

Paper

Code

Webpage

https://erizmr.github.io/UM2N/