High-order Discontinuous Galerkin Methods and Deep Reinforcement Learning with Application to Multi-scale Ocean Modeling

Corbin Foucart

PhD Defense, August 2, 2023

Committee members:

Prof. Pierre F.J. Lermusiaux[2,3]
Dr. Cuong Nguyen[1,2]
Prof. Anthony Patera[2,3]
Prof. Nicholas Patrikalakis[3]
Prof. Jaime Peraire[1,2]

Affiliations: [1] AeroAstro [2] CCSE [3] MechE

Outline

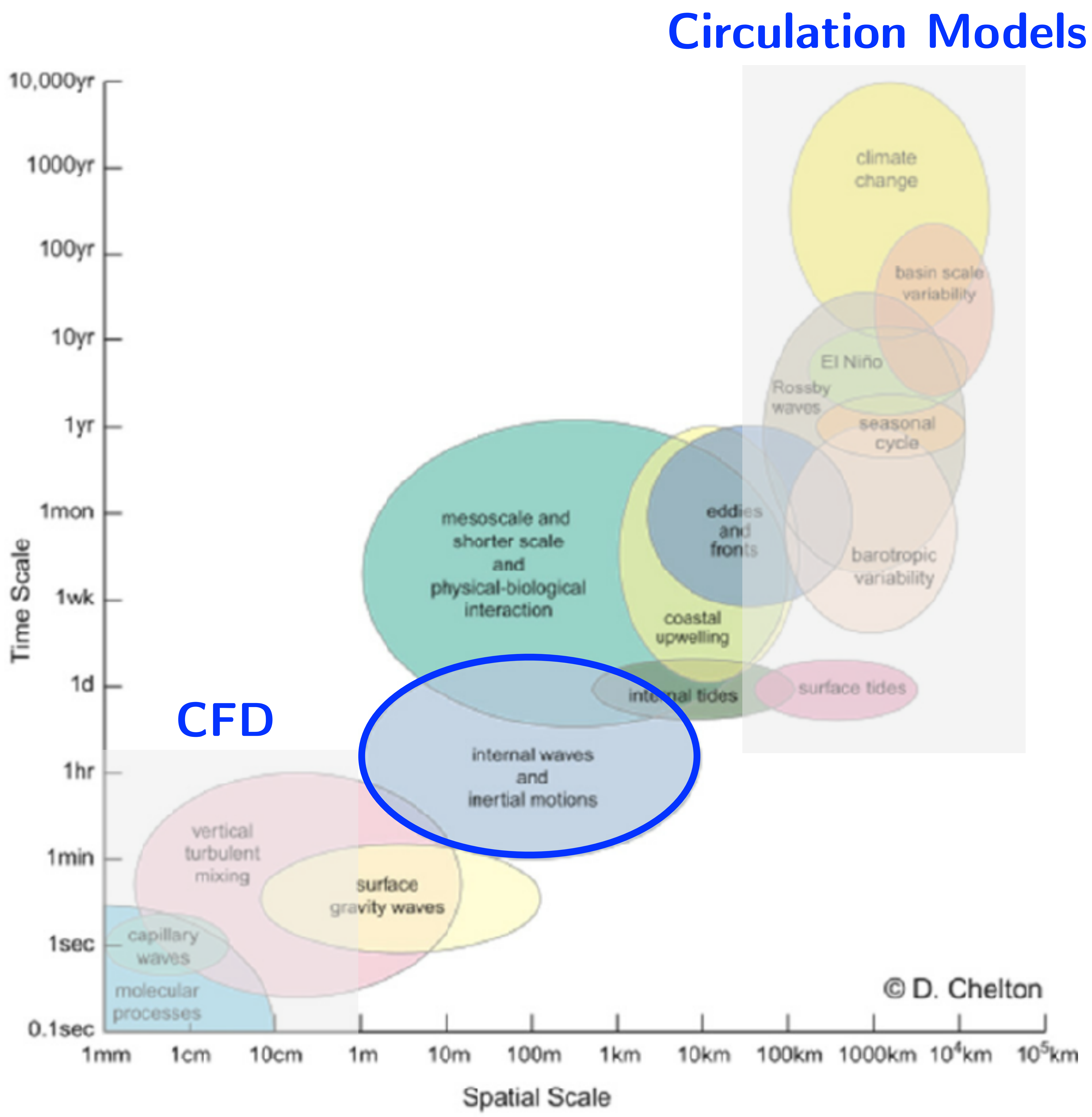
Need for nonhydrostatic ocean models

Importance of nonhydrostatic dynamics:

"As horizontal resolutions increase, nonhydrostatic effects become increasingly strong and presently are detectable at the submesoscale [1-5] and in deep convection [6] **...**

...few ocean circulation models presently have nonhydrostatic capability."

[Fox-kemper et al., *Challenges and prospects in ocean circulation models*, 2019]

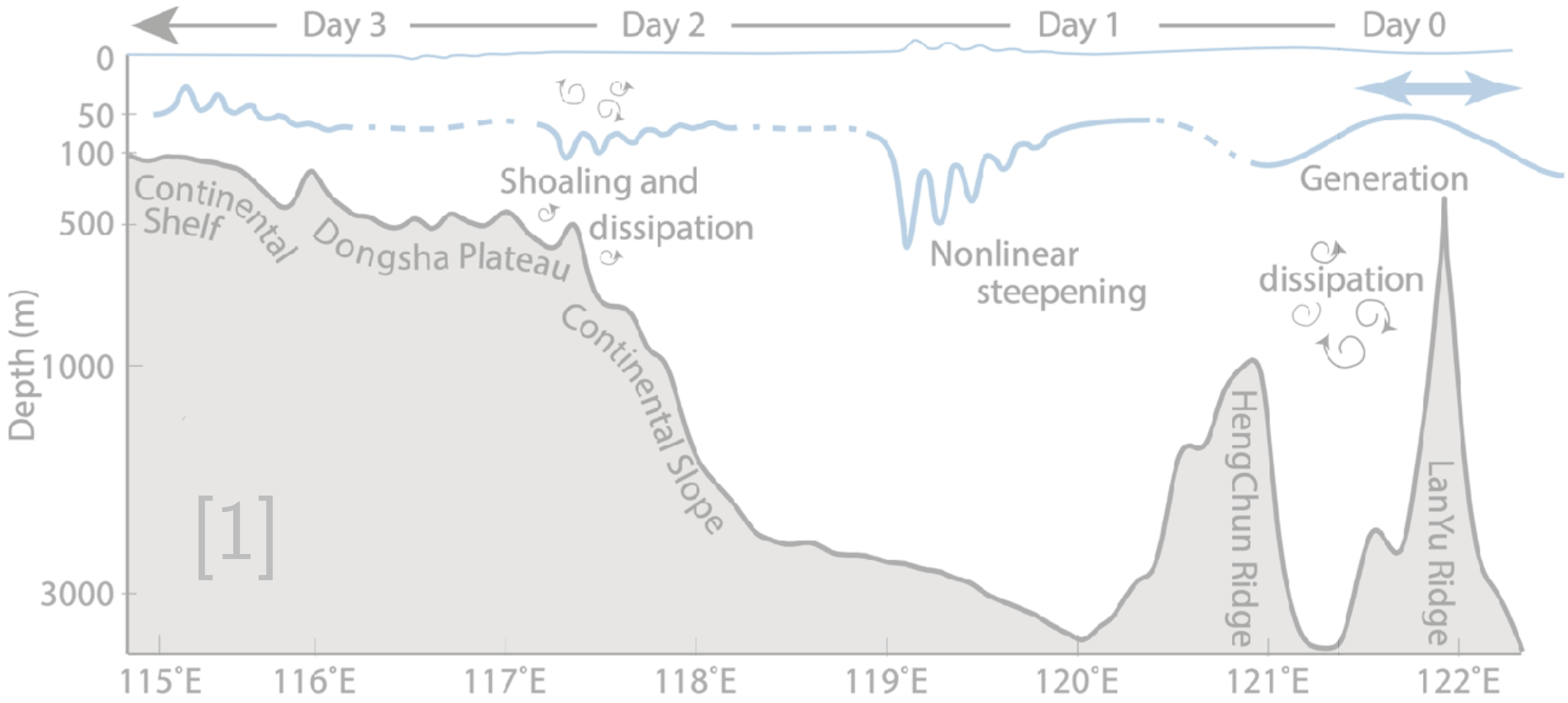


Need for nonhydrostatic ocean models

Internal waves:

- Disturbances that propagate in the interior of a stratified fluid [1]

propagation distance: $\mathcal{O}(1000\text{ km})$
wavelengths: $\mathcal{O}(1\text{ km})$

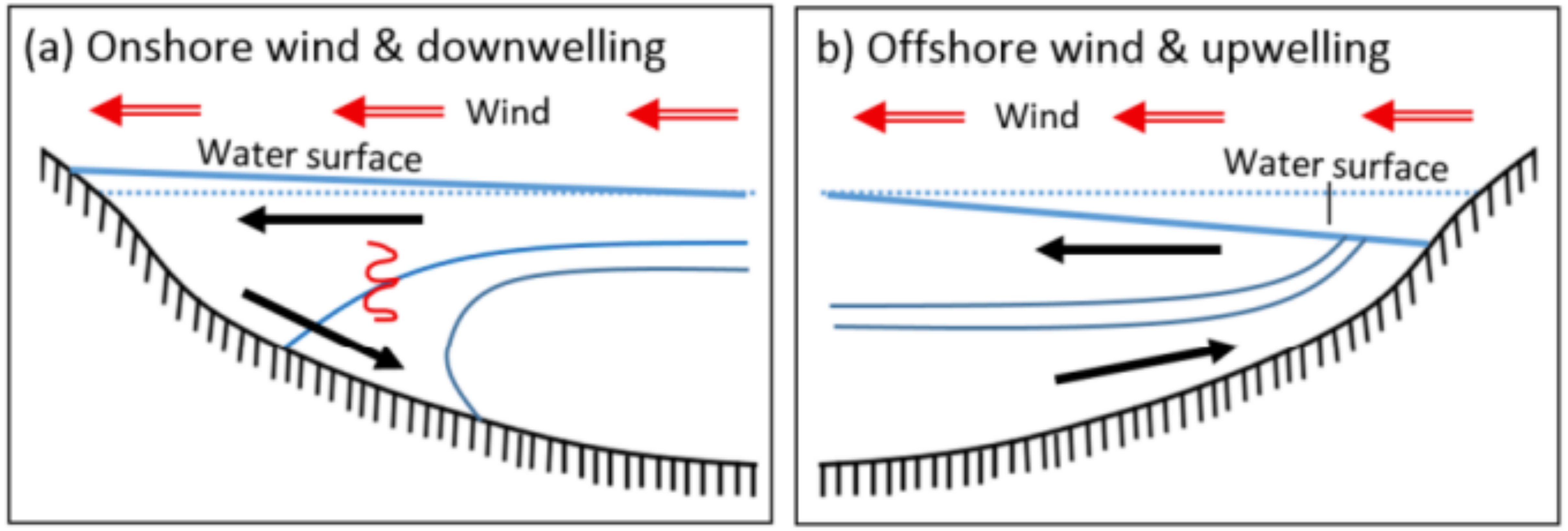


Need for nonhydrostatic ocean models

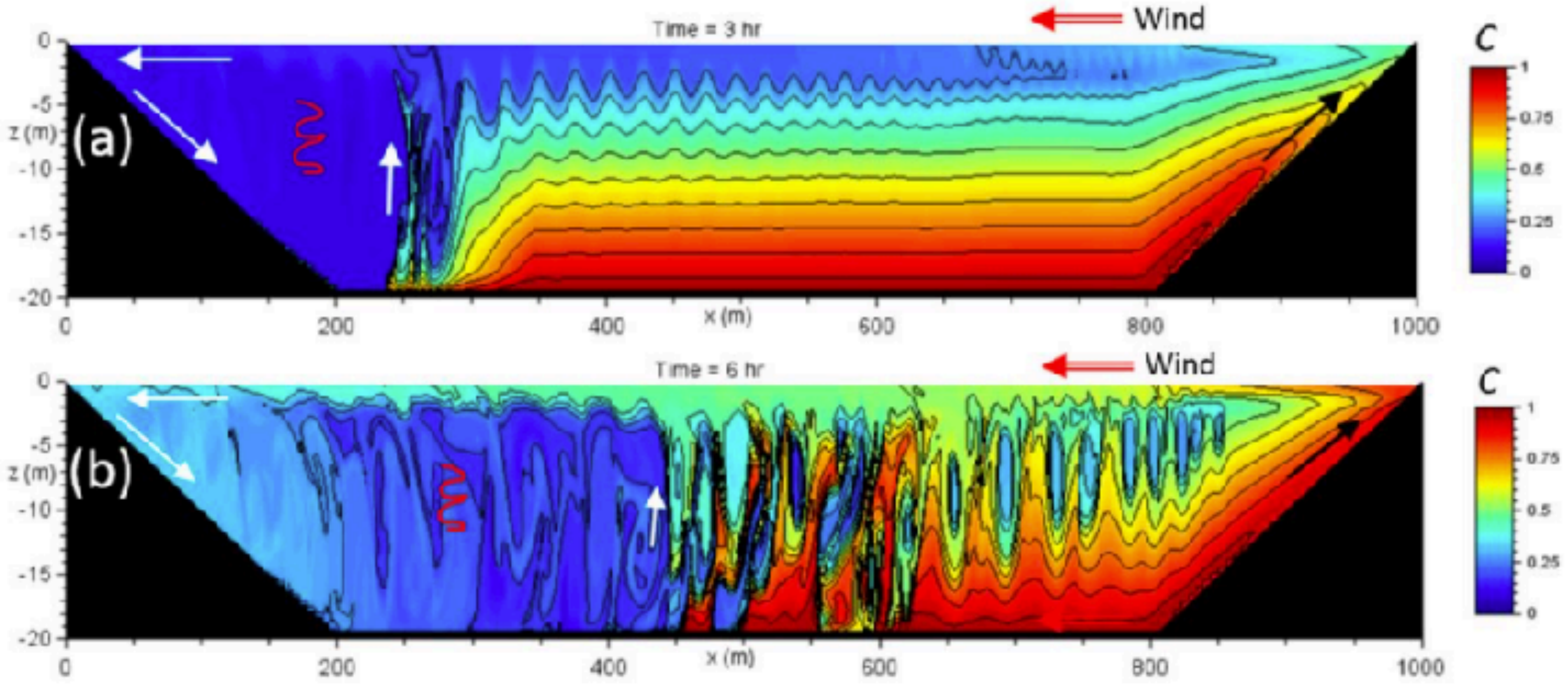
Subduction dynamics:

- Mechanisms of subduction are nonhydrostatic and not captured by regional models [2, 3] global models [1]
- Wind-driven surface stress can excite a dramatic dynamical response in the water column in terms of large-amplitude internal waves and strong vertical mixing [4, 5, 6]

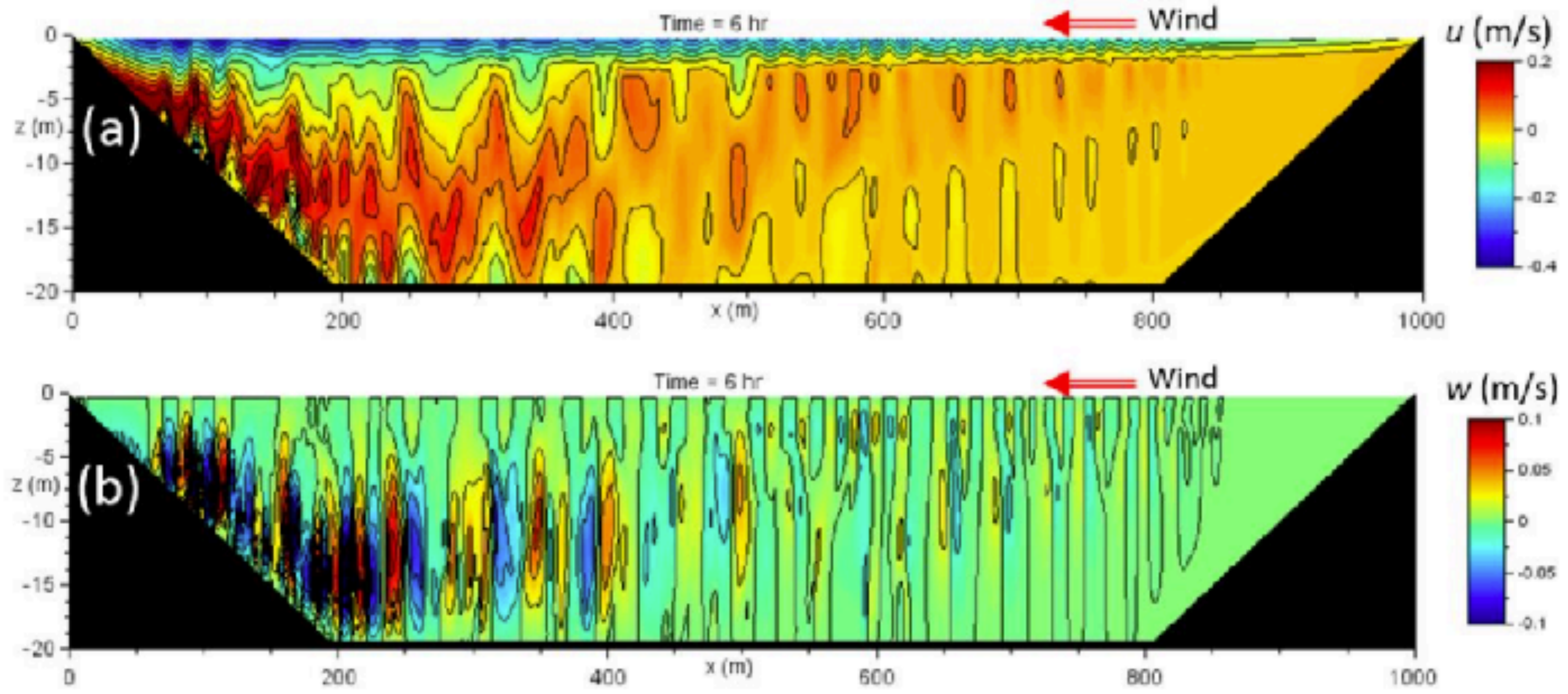
Seiche-like free-surface dynamics



Density



Velocity



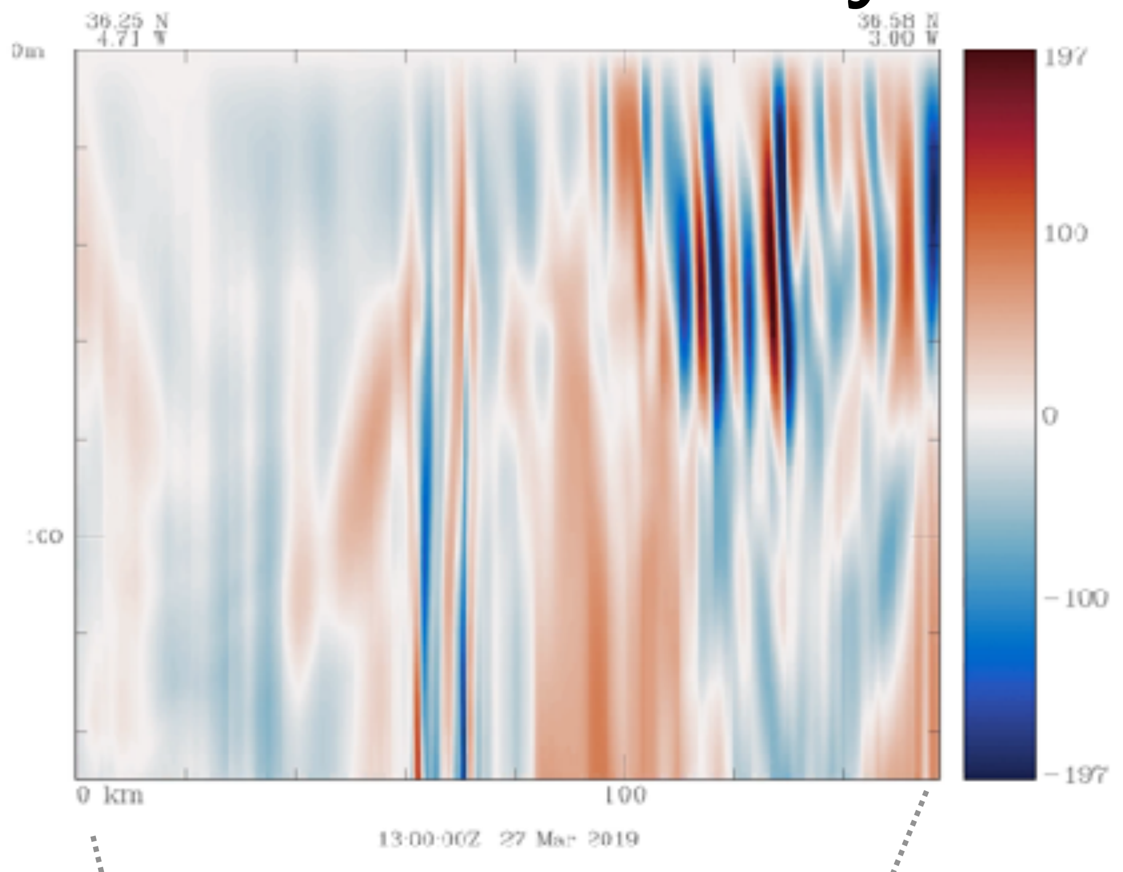
[1] Freilich & Mahadevan, JGR, 2021 [2] Pinardi et al, OM-2017 [3] Lermusiaux et al, OD-2013 [4] Calil, OD, 2017 [5] Kaempf, OD, 2017 [6] Kaempf, OD, 2019

Need for nonhydrostatic ocean models

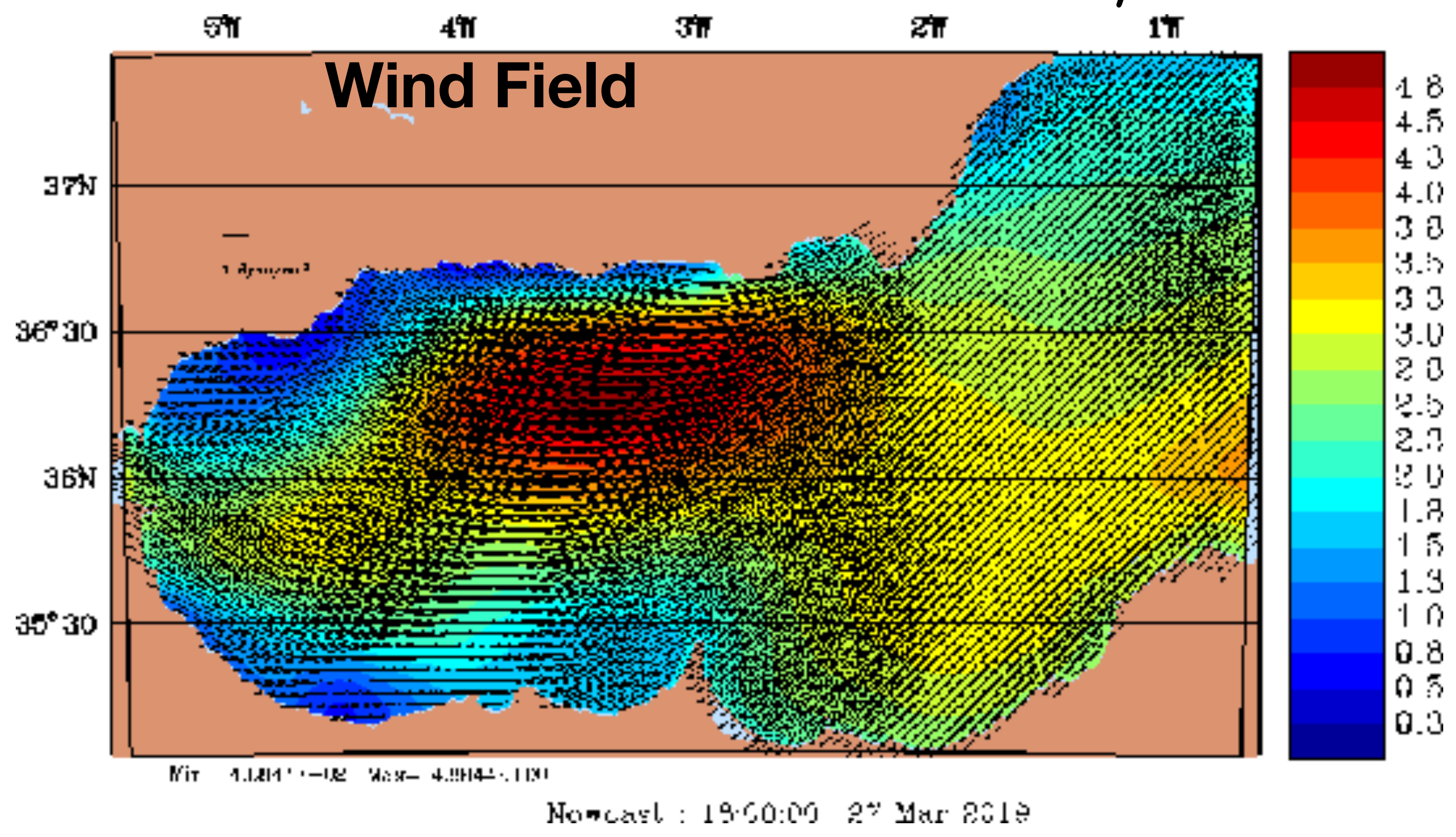
Subduction dynamics:

- Mechanisms of subduction are nonhydrostatic and not captured by regional models [2, 3] global models [1]
- Wind-driven surface stress can excite a dramatic dynamical response in the water column in terms of large-amplitude internal waves and strong vertical mixing [4, 5, 6]
- Frontal subduction, mixed-layer instabilities in Belearic and Alboran seas [7]
- Dynamics important for ocean acoustics, nutrient transport, carbon subduction from the atmosphere at small horizontal length scales [1-7]

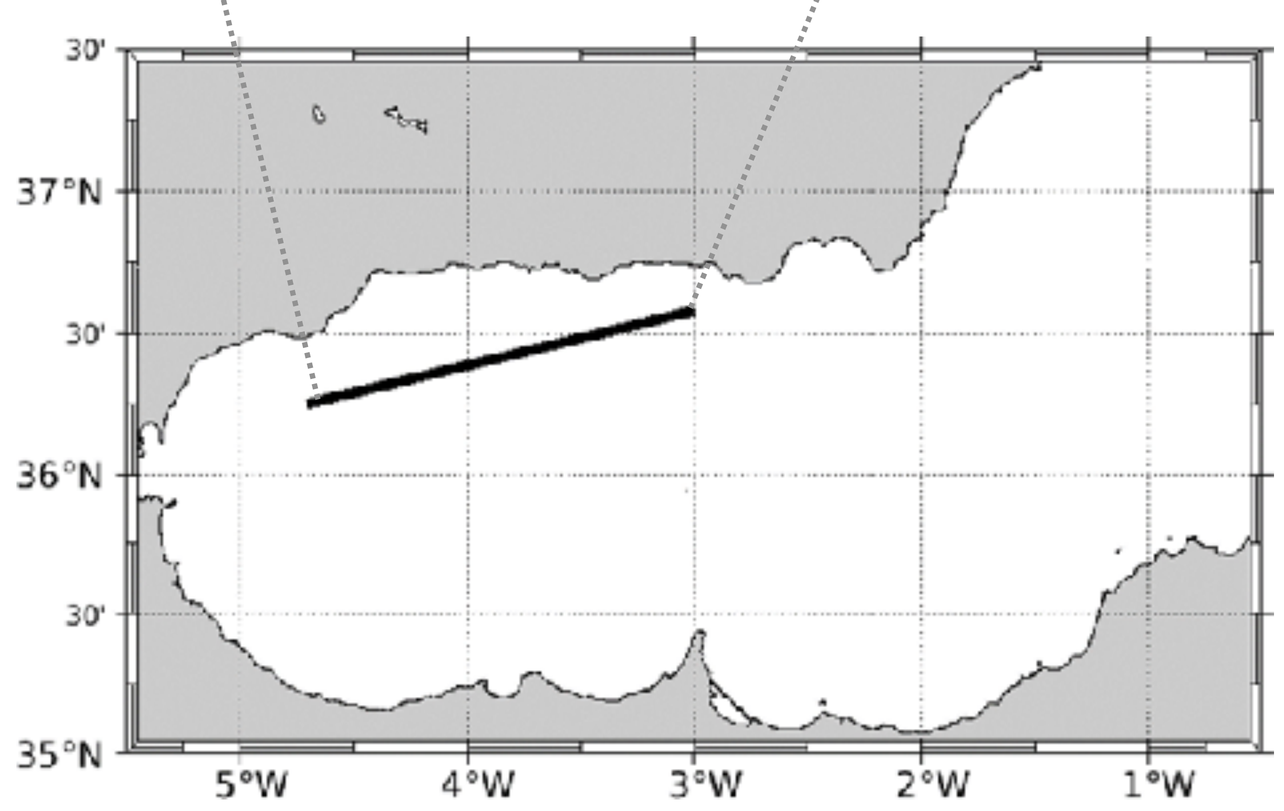
Vertical Velocity



Gale: March 27 00:00Z, W/SW



Modeling Domain [4]

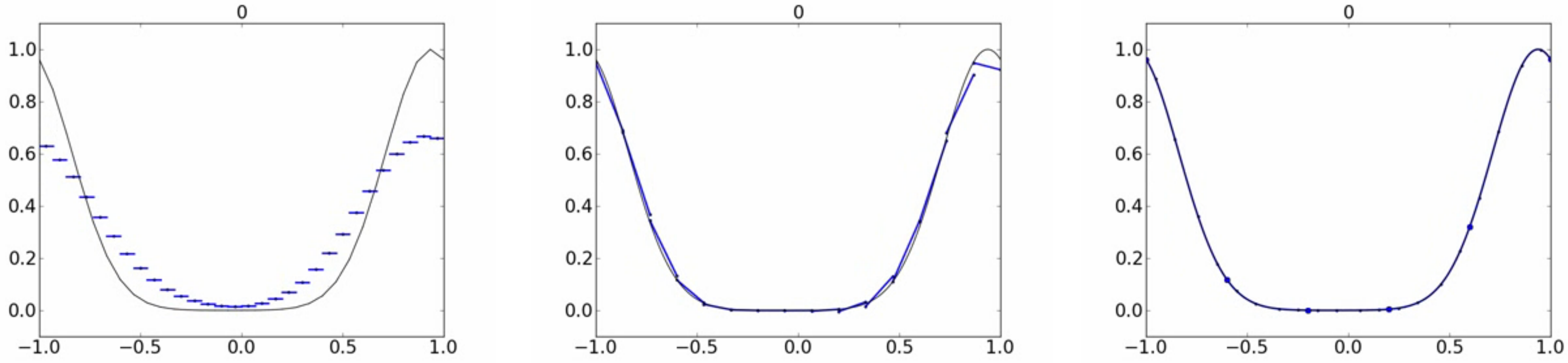


Why high-order methods?

Scalar advection equation: $\dfrac{\partial \phi}{\partial t} + \nabla \cdot (\mathbf{c}\phi) = 0$

All three simulations: same computational cost [2]

low-order | **high-order**



polynomial degree $p = 0$
30 elements

polynomial degree $p = 1$
15 elements

polynomial degree $p = 5$
5 elements

— Exact solution

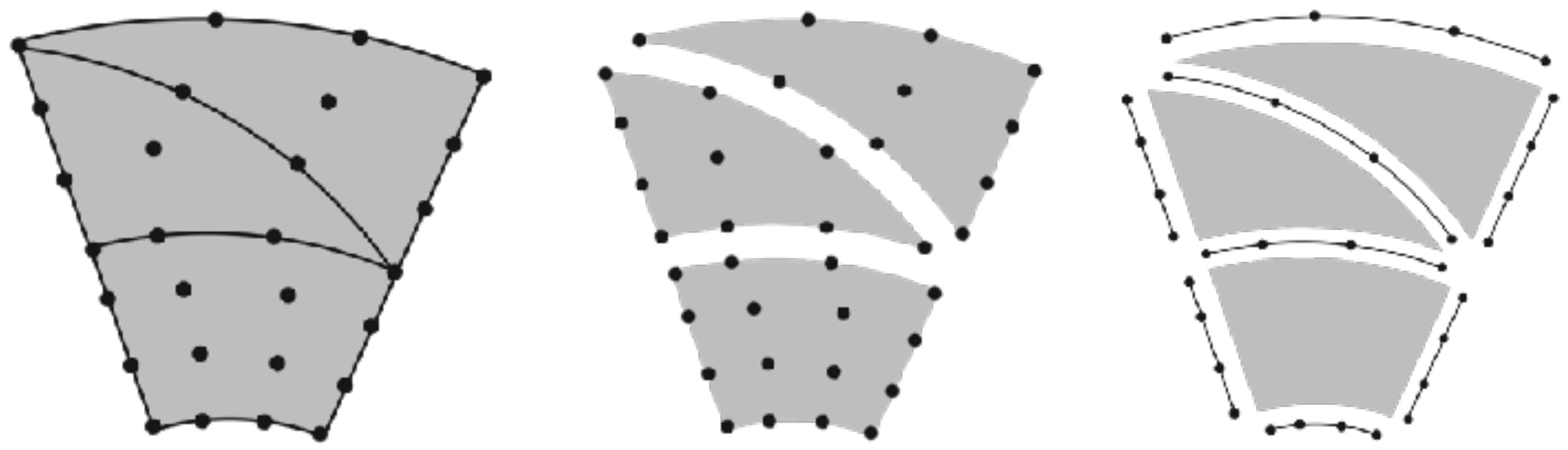
— Numerical solution

High-order methods often provide **more accurate solutions** for the **same computational cost** [1, 2]

Effectiveness of high-order DG methods

DG methods: incredibly successful modeling a wide range of phenomena in computational physics [1, 2]

Importantly: effective in *under-resolved* regimes



- Stability and robustness for projection, fully-implicit schemes [3]
- Able to transport high-frequency features over long time-integration horizons without significant dissipation/dispersion [4]
- Efficient on modern computer architectures [5, 6]
- Well-suited to adaptive mesh refinement [2]

State-of-the-art, incompressible Navier-Stokes equations

[1] Hesthaven 2008 [2] Kronbichler and Persson, Springer, 2021a [3] Fehn et al., JCP, 2017 [4] Fehn et al., JCP, 2018a [5] Fehn et al., JCP, 2018b [6] Kronbichler and Wall, SIAM, 2018 [7] Nguyen et. al, JCP, 2009a

Apply high-order DG methods to nonhydrostatic modeling

Ocean models: few have nonhydrostatic capability, and **low-order**

- PSOM (Mahadevan, 2020)
- MERF v3.0 (Tang et al., 2021)
- SUNTANS, GVC (Fringer et. al 2006; Fringer et al., 2011; Yue et al., 2021)
- MIT GCM (Marshall et al., 1997)—
- CROCO-ROMS (Cambon et al., 2018)
- Oceananigans.jl (Ramadhan et al., 2021)

State of the art, nonhydrostatic-capable models

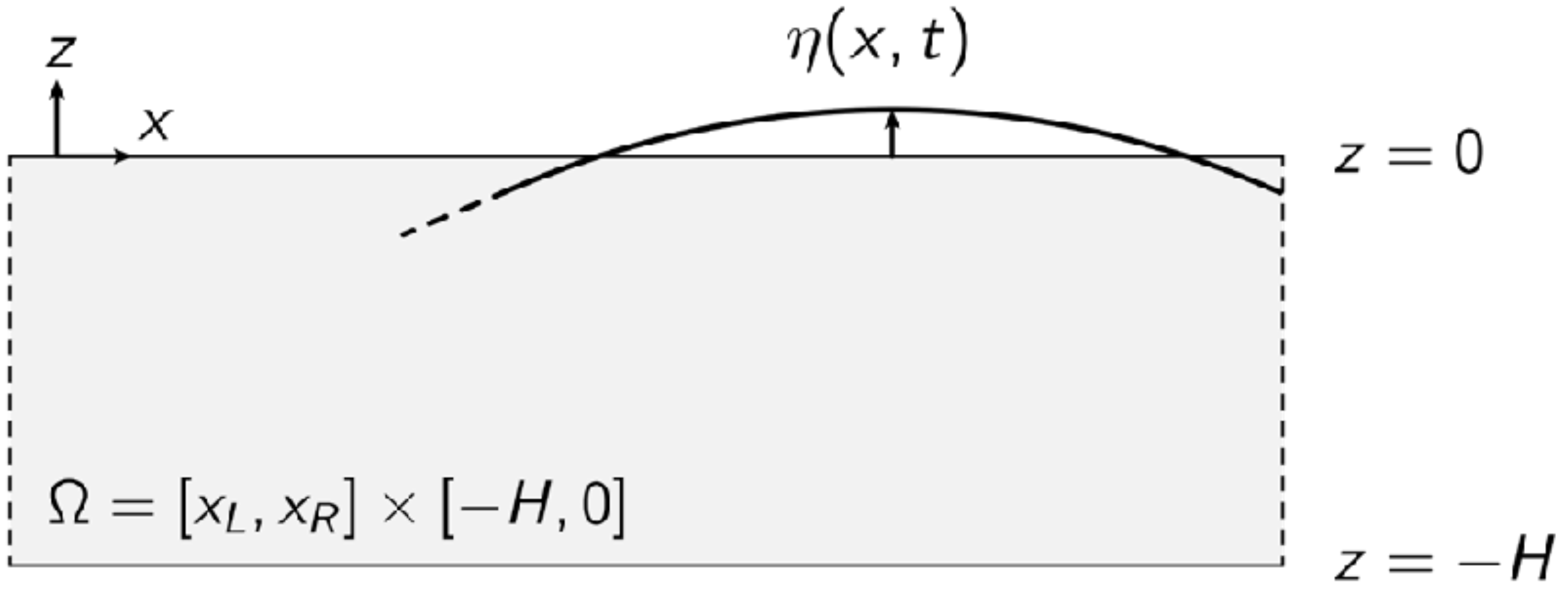
Use of high-order DG methods for nonhydrostatic modeling: promising, but still in its infancy [1-6]

[1] Ueckermann, 2014, PhD Thesis; [2] Ueckermann & Lermusiaux, 2016; [3] **Foucart** et. al, 2018; [4] Pan et. al, 2019,
[5] Pan et al., 2021, [6] **Foucart** et al., Ocean Modelling. in prep. 2023a.

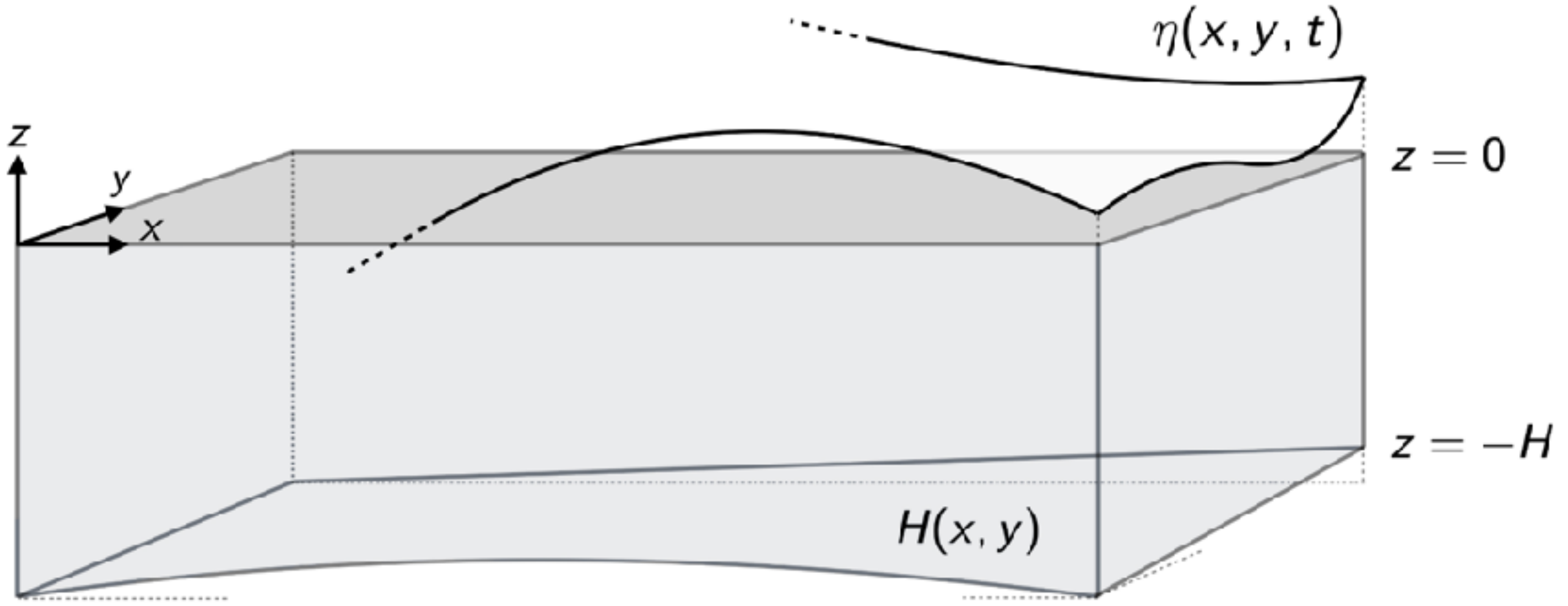
Outline

Model domain & geometry

2D



3D



Ocean Equations: temporal discretization [1, 2]

Ocean equations with a free-surface:

$$\frac{\partial \mathbf{u}}{\partial t} - (\nabla \cdot (\boldsymbol{\nu} \nabla \mathbf{u})) + \nabla p' + g \nabla_{xy} \eta = -\frac{1}{\rho_0} \int_z^{\eta} g \nabla_{xy} \rho' \, dz' - \nabla \cdot (\mathbf{u} \otimes \mathbf{u}) + \frac{1}{\rho_0} \mathbf{f}$$

$$\frac{\partial \eta}{\partial t} + \nabla \cdot \left(\int_{-H}^{\eta} \mathbf{u} \, dz \right) = 0$$

$$\nabla \cdot \mathbf{u} = 0$$

$$\frac{\partial \rho'}{\partial t} - \nabla \cdot (\kappa \nabla \rho') = -\nabla \cdot (\mathbf{u} \rho') + f_{\rho'}$$

where:

$$p = p_{\text{hyd}} + \rho_0 p', \qquad p_{\text{hyd}} = \int_z^{\eta} \rho g \, dz'.$$

ν, $\kappa \in \mathbb{R}^{d \times d}$ are diffusivity tensors

velocity predictor [3D]

$$\frac{\bar{\mathbf{u}}^{k+1}}{a\Delta t} - \left(\nabla \cdot \left(\boldsymbol{\nu} \nabla \bar{\mathbf{u}}^{k+1} \right) \right) + \nabla p'^{,k} + g \nabla_{xy} \eta^k = \mathbf{F}_{\bar{\mathbf{u}}}^{k,k+1}$$

free-surface correction [2D]

$$\frac{\delta \eta^{k+1}}{a\Delta t} - \nabla \cdot \left(a\Delta t g \left(\eta^k + H \right) \nabla \delta \eta^{k+1} \right) = -\nabla \cdot \int_{-H}^{\eta^k} \bar{\mathbf{u}}^{k+1} \, dz$$

$$\bar{\bar{\mathbf{u}}}_{xy}^{k+1} = \bar{\mathbf{u}}_{xy}^{k+1} - (a\Delta t) g \nabla_{xy} \delta \eta^{k+1}$$

$$\eta^{k+1} = \eta^k + \delta \eta^{k+1}$$

pressure correction [3D]

$$\nabla^2 \delta p'^{,k+1} = \frac{\nabla \cdot \bar{\bar{\mathbf{u}}}^{k+1}}{a\Delta t}$$

$$\mathbf{u}^{k+1} = \bar{\bar{\mathbf{u}}}^{k+1} - a\Delta t \nabla \delta p'^{,k+1},$$

$$p'^{,k+1} = p'^{,k} + \delta p'^{,k+1}.$$

tracer evolution [3D]

$$\frac{\rho'^{k+1}}{a\Delta t} - \nabla \cdot \left(\kappa \nabla \rho'^{k+1} \right) - \nabla \cdot \left(\mathbf{u}^{k+1} \rho'^{k+1} \right) = \frac{\rho'^{k}}{a\Delta t} + f_{\rho'}.$$

Contribution: novel spatial discretization of ocean equations

- Derived & implemented a new HDG spatial discretization of the free-surface ocean equations (nonhydrostatic and hydrostatic) [1, 2]
- Incorporated ideas from classical HDG literature [3, 4] as well as modern results from DG literature pertaining to robust projection schemes for under-resolved incompressible flows [5, 6, 7]
- Implemented the discretization in a parallelized C++ framework tailored to HDG schemes
  - originally implemented in Python, then re-written in C++ for performance
  - multi-threaded implementation—parallelized HDG assembly/reconstruction
  - template meta-programming for dimension-specific compiler optimizations
  - suite of ~200 tests that verify implementations, optimal convergence rates for all solvers

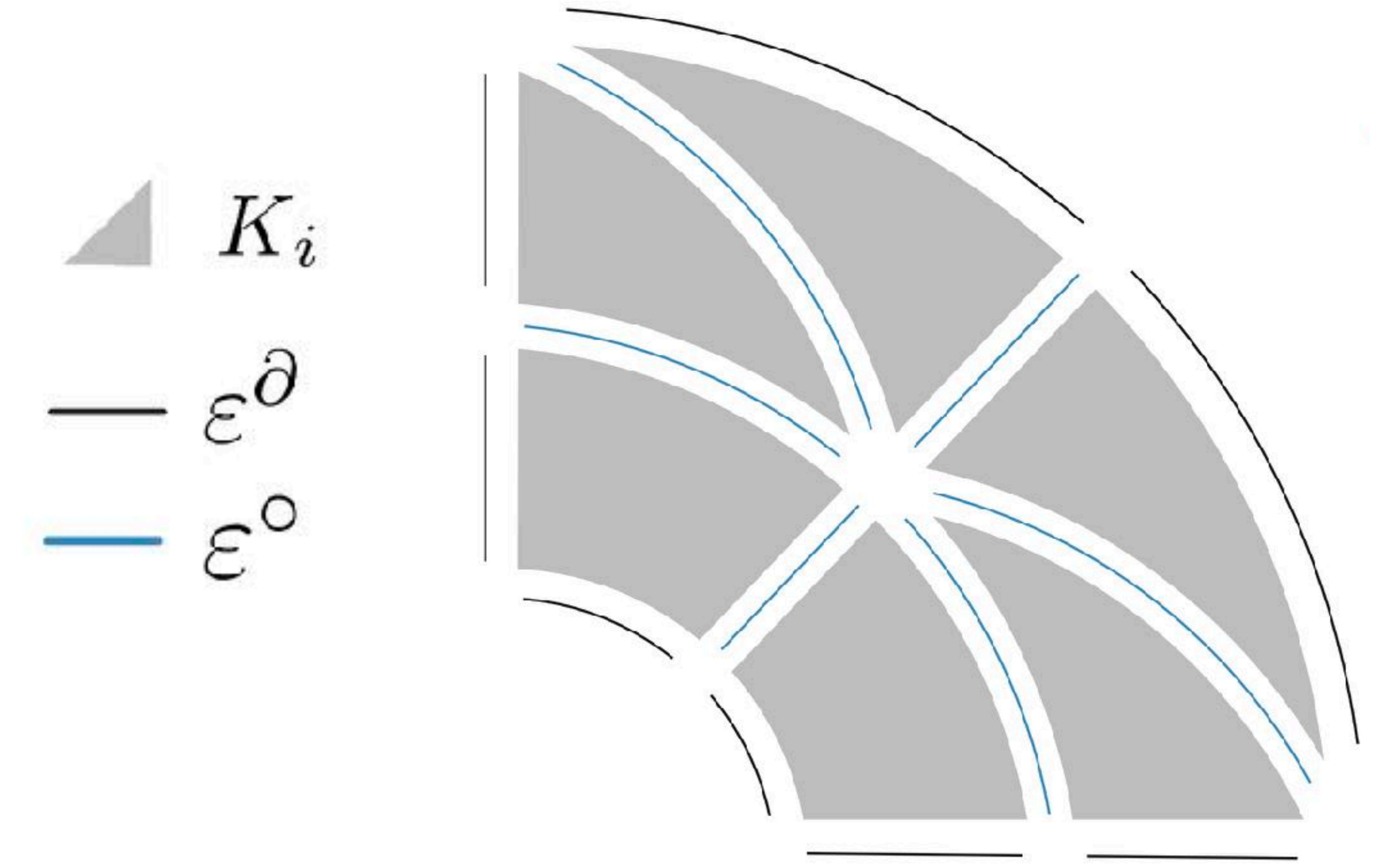
[1] **Foucart** et al., 2021. [2] **Foucart** et al., Ocean Modelling. in prep. 2023a. [3,4] Nguyen et al., JCP. 2009a,b. [5] Fehn et al., JCP. 2017. [6] Fehn et al., JCP. 2018. [7] Kronbichler & Persson, Springer, 2021.

Spatial discretization: notation, finite element spaces

$\underline{d = 2}$

Mesh:

- Integer d: spatial dimension
- $\mathcal{T}_h = \cup_i K_i$: a finite collection of non-overlapping elements K_i discretizing $\Omega \subset \mathbb{R}^d$, with boundary Γ
- $\partial\mathcal{T}_h = \{\partial K : K \in \mathcal{T}_h\}$ interfaces and boundary edges of the elements, where ∂K is the boundary of element K
- Each edge $e \in \varepsilon^\circ$ or $e \in \varepsilon^\partial$, (interior and boundary edges)



Finite element spaces:

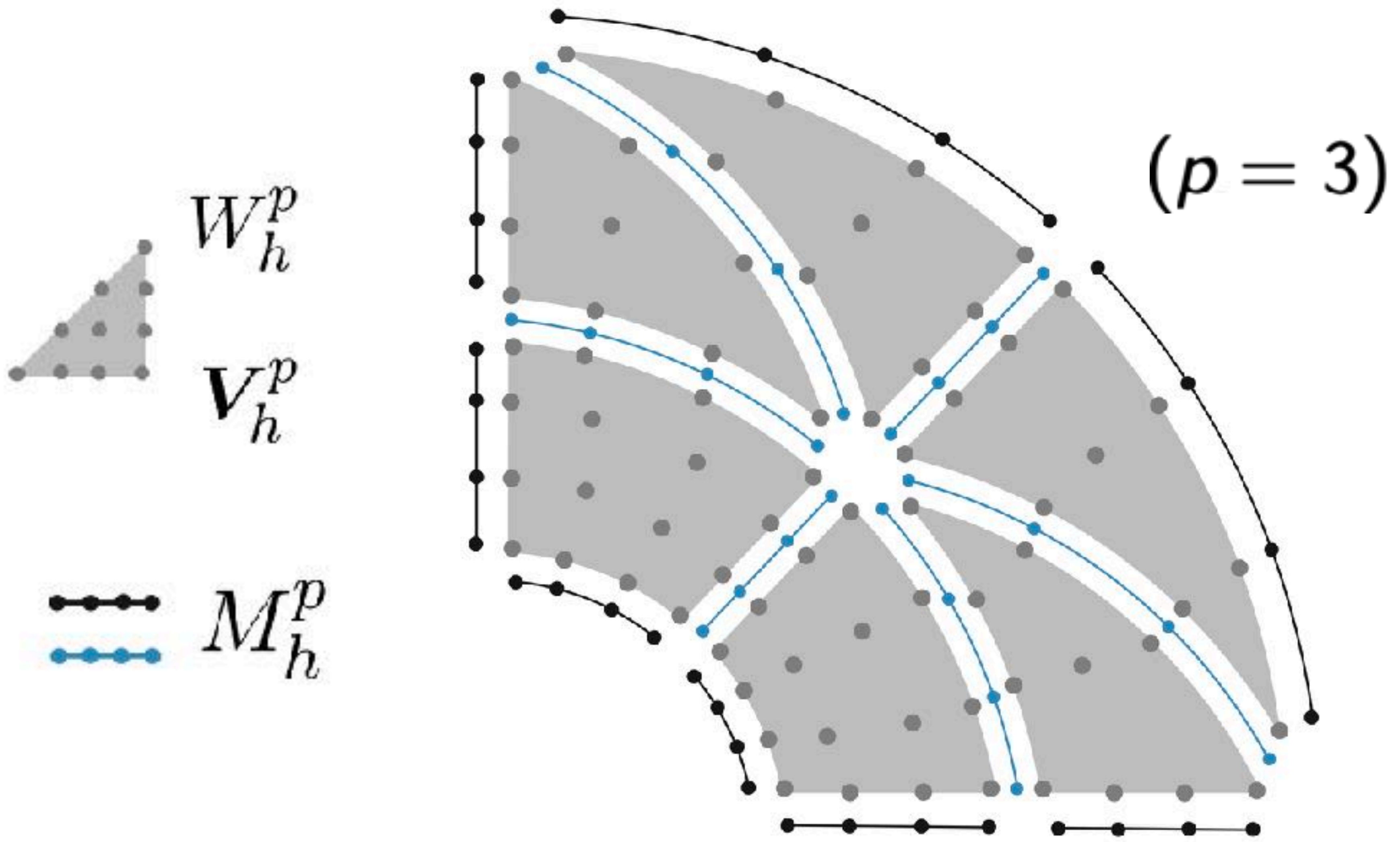
Let $\mathcal{P}^p(D)$ denote the set of polynomials of degree p on a domain D.

$$\mathbf{G}_h^p = \left\{\mathbf{G} \in \left[L^2(\Omega)\right]^{d\times d} : \mathbf{G}\big|_K \in [\mathcal{P}^p(K)]^{d\times d}\, \forall K \in \mathcal{T}_h\right\},$$

$$\mathbf{V}_h^p = \left\{\mathbf{v} \in \left[L^2(\Omega)\right]^{d} : \mathbf{v}\big|_K \in [\mathcal{P}^p(K)]^{d}\, \forall K \in \mathcal{T}_h\right\},$$

$$W_h^p = \left\{w \in L^2(\Omega) : w\big|_K \in \mathcal{P}^p(K)\, \forall K \in \mathcal{T}_h\right\},$$

$$M_h^p = \left\{\mu \in L^2(\varepsilon_h) : \mu\big|_e \in \mathcal{P}^p(e)\, \forall e \in \varepsilon_h\right\},$$



Weak formulation: velocity predictor

find: $(\bar{\mathbf{u}}_h^{k+1}, \bar{\mathbf{L}}_h, \hat{\mathbf{u}}_h) \in (\mathbf{G}_h \times \mathbf{V}_h \times \mathbf{M}_h)$ such that

$$(\mathbf{G}, \bar{\mathbf{L}}_h)_{\mathcal{T}_h} + (\nabla \cdot \mathbf{G}, \bar{\mathbf{u}}_h^{k+1})_{\mathcal{T}_h} - \langle \mathbf{G} \cdot \mathbf{n}, \hat{\mathbf{u}}_h \rangle_{\partial \mathcal{T}_h} = 0$$

$$\left(\mathbf{v}, \frac{\bar{\mathbf{u}}^{k+1}}{a\Delta t}\right)_{\mathcal{T}_h} - \left(\mathbf{v}, \nabla \cdot (\nu \bar{\mathbf{L}}_h)\right)_{\mathcal{T}_h} + \langle \mathbf{v}, \tau\left(\bar{\mathbf{u}}_h^{k+1} - \hat{\mathbf{u}}_h\right)\rangle_{\partial \mathcal{T}_h}$$
$$= -a_h(\mathbf{u}^k, \mathbf{g}_D) - \mathrm{pg}_h\left(\mathbf{v}, p_h^k\right) - (\mathbf{v}, \mathbf{F}_{\rho'} + \mathbf{F}_\eta + \mathbf{F}_{\mathrm{cor}})_{\mathcal{T}_h} + (\mathbf{v}, \mathbf{F}_t)_{\mathcal{T}_h}$$

$$\langle \boldsymbol{\mu}, (-\nu \bar{\mathbf{L}}_h)\mathbf{n} + \tau(\bar{\mathbf{u}}_h^{k+1} - \hat{\mathbf{u}}_h)\rangle_{\partial \mathcal{T}_h \setminus \Gamma_D} + \left\langle \boldsymbol{\mu}, \widehat{b}_h \right\rangle_{\Gamma_N} = 0$$

for all $(\mathbf{G}, \mathbf{v}, \boldsymbol{\mu}) \in (\mathbf{G}_h \times \mathbf{V}_h \times \mathbf{M}_h)$.

Body forcing terms: $\mathbf{F}_{\rho'} = 1/\rho_0 \int_z^{\eta^k} g \nabla_{xy} \rho'^{,k}\, dz'$, $\mathbf{F}_\eta = g \nabla_{xy} \eta^k$, $\mathbf{F}_{\mathrm{cor}} = f_c \hat{z} \times \mathbf{u}^k$, $\mathbf{F}_t = \frac{1}{a\Delta t}\mathbf{u}^k$,

Advection term:

$$a_h(\mathbf{v}, \mathbf{u}_h^k, \mathbf{g}_D) = -\left(\nabla \mathbf{v}, \mathbf{F}_a(\mathbf{u}_h^k)\right)_K + \left\langle \mathbf{v}, \mathbf{F}_a^*(\mathbf{u}_h^k, \mathbf{g}_D)\right\rangle_{\partial K},$$

Nonhydrostatic pressure gradient term:

$$\mathrm{pg}_h(\mathbf{v}, p_h^k) = -\left(\nabla \cdot \mathbf{v}, p_h'^{,k}\right)_K + \langle \mathbf{v}, (p_h')^* \mathbf{n}\rangle_{\partial K},$$

Numerical flux definitions:

$$\mathbf{F}_a^*(\mathbf{u}_h) = \mathbf{F}_a(\mathbf{u}_h^k)_{\mathrm{upwind}}$$

$$(p_h')^* = \{\{p_h'\}\}$$

Weak formulation: free-surface correction (dim - 1)

Define the gradient $\mathbf{q}_{\delta\eta} \equiv \nabla \delta\eta$. $\quad \bar{\mathbf{U}}_h \equiv \int_{-H}^{\eta^k} \bar{\mathbf{u}}_{xy,h}^{k+1}\, dz$

find: $(\delta\eta_h^{k+1}, \mathbf{q}_{h,\delta\eta}, \hat{\delta\eta}_h) \in (W_h \times \mathbf{V}_h \times M_h)$ such that

$$\left(\mathbf{v}, \frac{\mathbf{q}_{h,\delta\eta}}{a\Delta t g\left[\eta^k + H\right]}\right)_{\mathcal{T}_h} + \left(\nabla \cdot \mathbf{v}, \delta\eta_h^{k+1}\right)_{\mathcal{T}_h} - \left\langle \mathbf{v} \cdot \mathbf{n}, \hat{\delta\eta}_h \right\rangle_{\partial\mathcal{T}_h} = 0,$$

$$\left(w, \frac{\delta\eta_h^{k+1}}{a\Delta t}\right)_{\mathcal{T}_h} - \left(w, \nabla \cdot \mathbf{q}_{h,\delta\eta}\right)_{\mathcal{T}_h} + \left\langle w, \tau\left(\delta\eta_h^{k+1} - \hat{\delta\eta}_h\right)\right\rangle_{\partial\mathcal{T}_h} = -d_h(w, \bar{\mathbf{U}}_h),$$

$$\left\langle \mu, \mathbf{q}_{h,\delta\eta} \cdot \mathbf{n} + \tau\left(\delta\eta_h^{k+1} - \hat{\delta\eta}_h\right)\right\rangle_{\partial\mathcal{T}_h} = \langle \mu, g_N\rangle_{\partial\mathcal{T}_h}$$

for all $(w, v, \mu) \in (W_h \times \mathbf{V}_h \times M_h)$

Discrete depth-integrated divergence

$$d_h(w, \bar{\mathbf{U}}) = \left(w, \nabla \cdot \bar{\mathbf{U}}\right)_K = -\left(\nabla w, \bar{\mathbf{U}}\right)_K + \left\langle w, \left(\bar{\mathbf{U}}\right)^* \cdot \mathbf{n}\right\rangle_{\partial K}$$

Numerical flux definition:

$$\left(\bar{\mathbf{U}}\right)^* = \{\{\bar{\mathbf{U}}\}\}$$

Weak formulation: pressure correction

Define the gradient of the pressure corrector $\mathbf{q}_{\delta p'} = \nabla \delta p'$.

find: $(\mathbf{q}_{\delta p'}, \delta p', \hat{\delta p'}) \in (\mathbf{V}_h \times W_h \times M_h)$ such that

$$\left(\mathbf{v},\, \mathbf{q}_{\delta p'}^{k+1}\right)_{\mathcal{T}_h} + \left(\nabla \cdot \mathbf{v},\, \delta p'^{,k+1}\right)_{\mathcal{T}_h} - \left\langle \mathbf{v} \cdot \mathbf{n},\, \widehat{\delta p'} \right\rangle_{\partial \mathcal{T}_h} = 0,$$

$$-\left(w, \nabla \cdot \mathbf{q}_{\delta p'}^{k+1}\right)_{\mathcal{T}_h} + \left\langle w,\, \tau_p\, \delta p'^{,k+1} \right\rangle_{\partial \mathcal{T}_h} - \left\langle w,\, \tau_p \widehat{\delta p'} \right\rangle_{\partial \mathcal{T}_h} = \frac{1}{a\Delta t}\left[-\left(\nabla w,\, \bar{\bar{\mathbf{u}}}^{k+1}\right)_{\mathcal{T}_h} + \left\langle w,\, \left(\bar{\bar{\mathbf{u}}}^{k+1}\right)^* \cdot \mathbf{n} \right\rangle_{\partial \mathcal{T}_h}\right],$$

$$\left\langle \mu,\, \mathbf{q}_{\delta p'}^{k+1} \cdot \mathbf{n} + \tau_p \left(\delta p'^{,k+1} - \hat{\delta p'}\right) \right\rangle_{\delta \mathcal{T}_h \backslash \Gamma_d} = 0$$

for all $(w, v, \mu) \in (W_h \times \mathbf{V}_h \times M_h)$

Numerical flux definition:

$$\left(\bar{\bar{\mathbf{u}}}^{k+1}\right)^* = \{\{\bar{\bar{\mathbf{u}}}^{k+1}\}\}$$

Verification: convergence

- Spatial convergence test: method of manufactured solutions, inspired by [2]. For each weak form (in 2D, 3D):

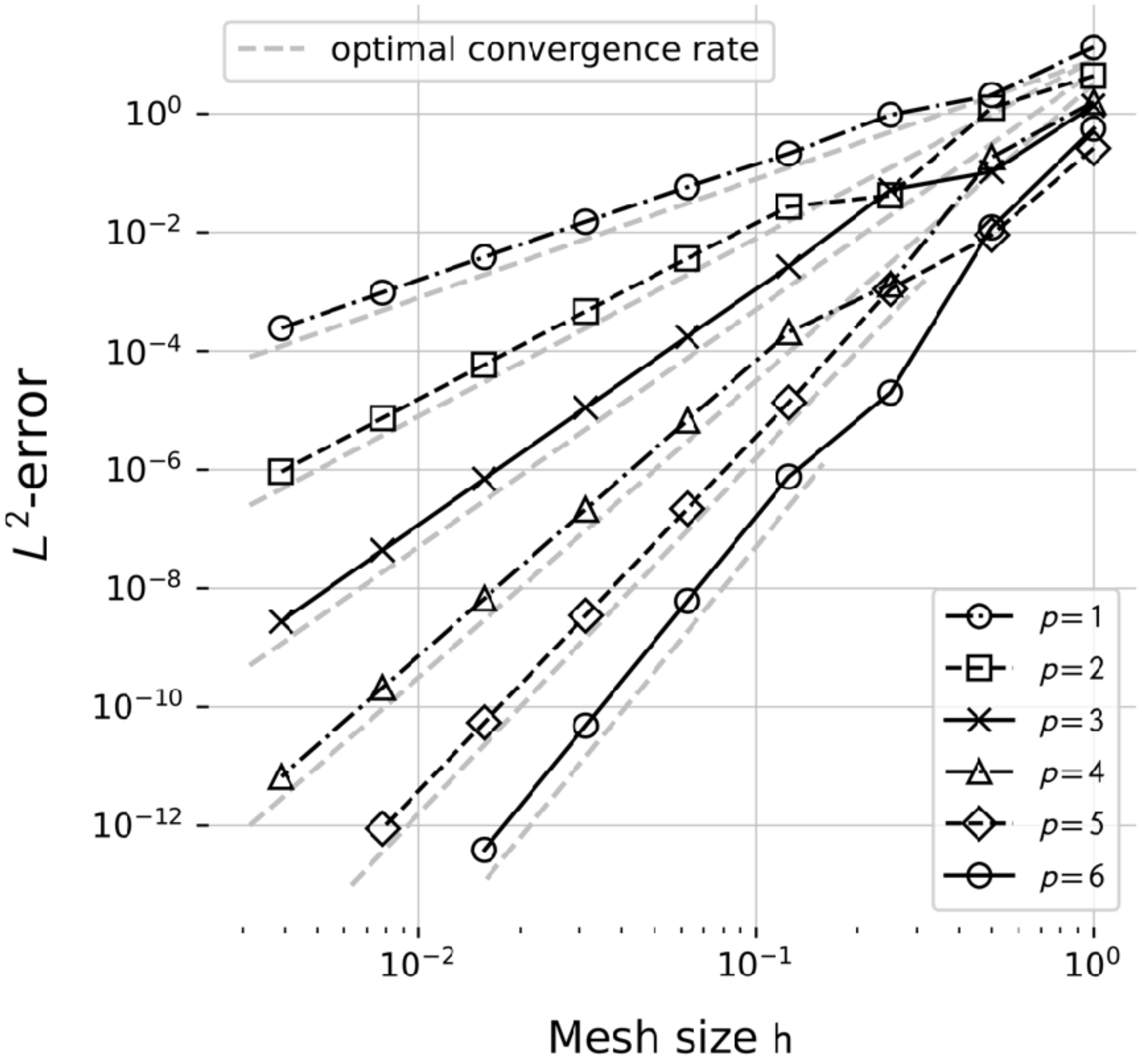
  choose $\Upsilon = (\mathbf{u}, p', \eta, \rho')$

  deduce the forcing term $\mathbf{f}$

  measure L^2-error as (regular) mesh is refined

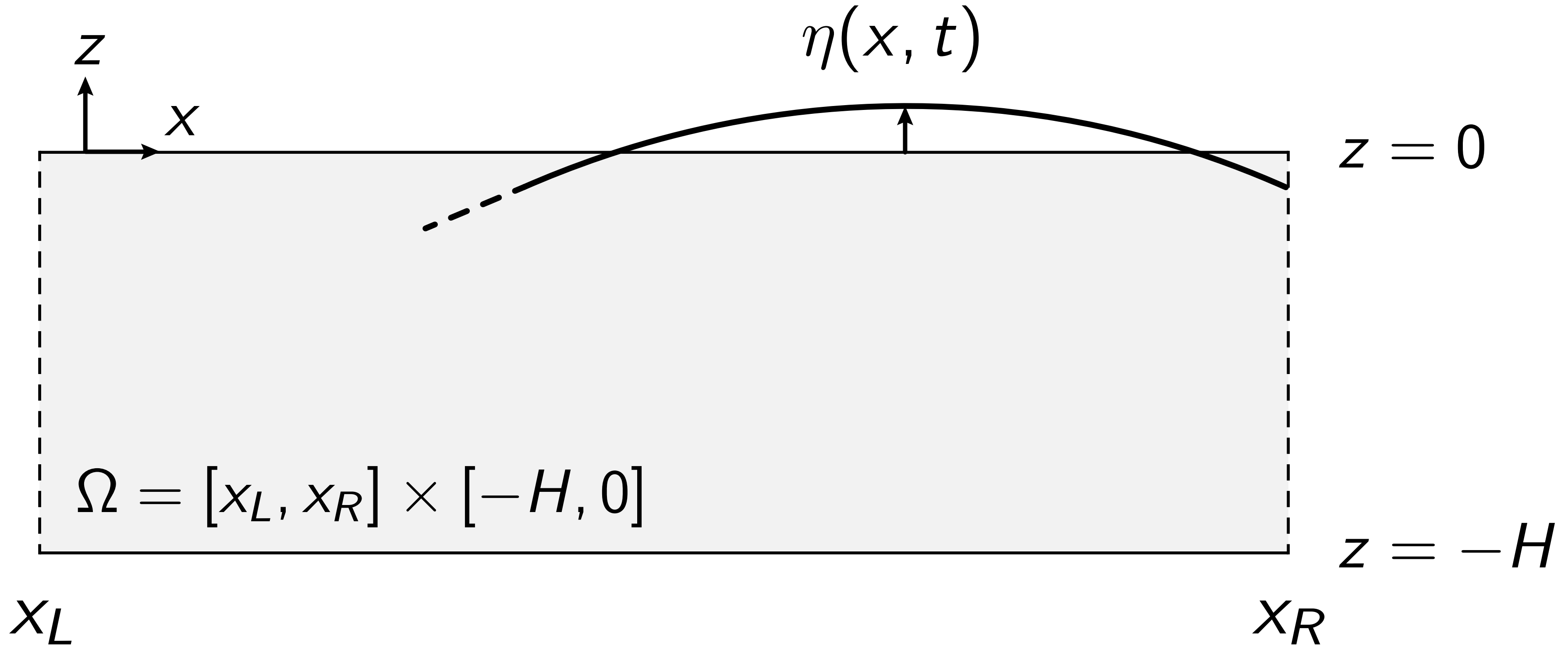
- Optimal p+1 convergence observed in both primal and gradient unknowns (HDG) (shown for the momentum equation, right)

[1]



Verification of NHS model: free-surface seiche

Modeling Domain, Boundary Conditions

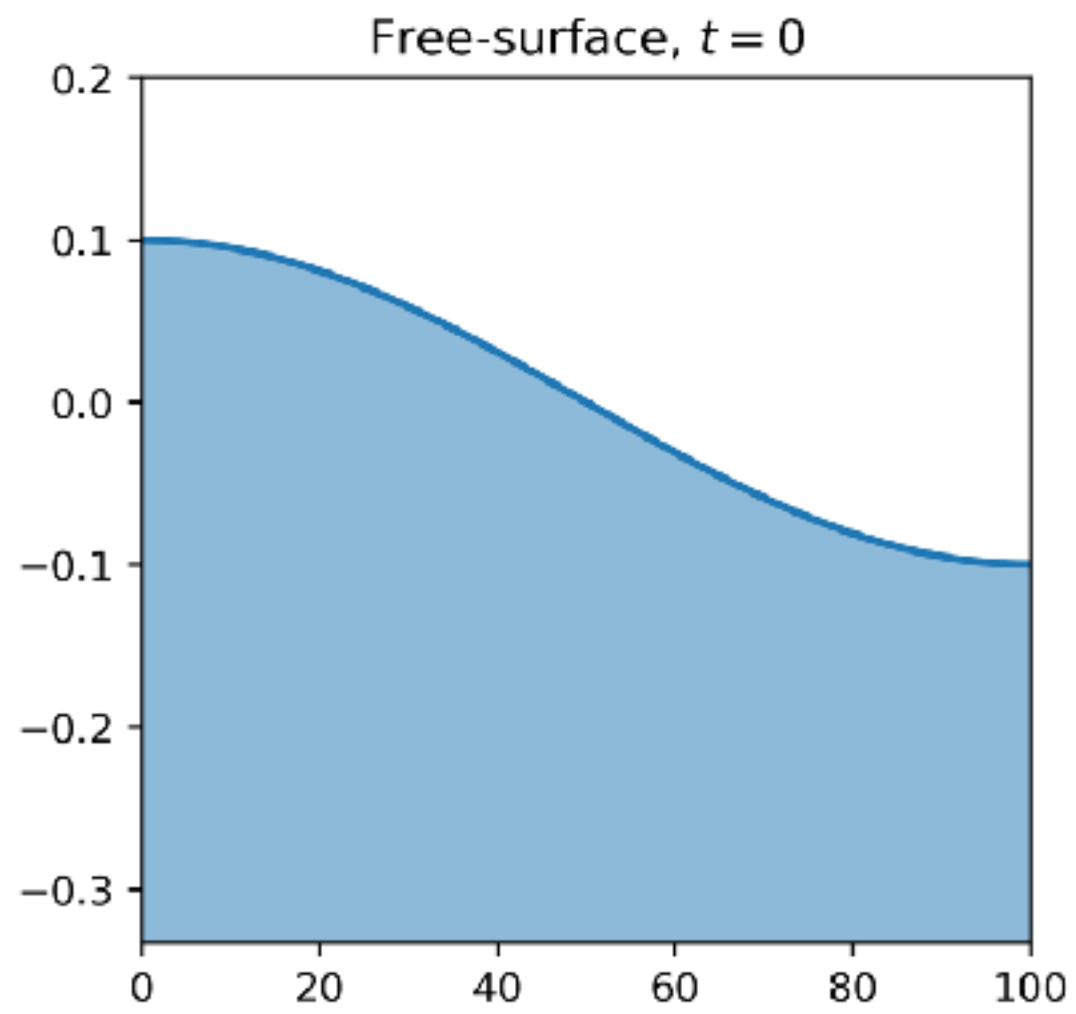


	top	bottom	sides
$\bar{u}$	Γ_N	Γ_N	Γ_D
$\bar{w}$	Γ_N	Γ_D	Γ_N
$\delta\eta$	-	-	Γ_N
$\delta p'$	Γ_D	Γ_N	Γ_N
ρ'	Γ_N	Γ_N	Γ_N

Initial Condition:

$$\eta(t=0) = a\cos(kx)$$

$$k = \frac{\pi}{L}$$



Verification of NHS model: free-surface seiche

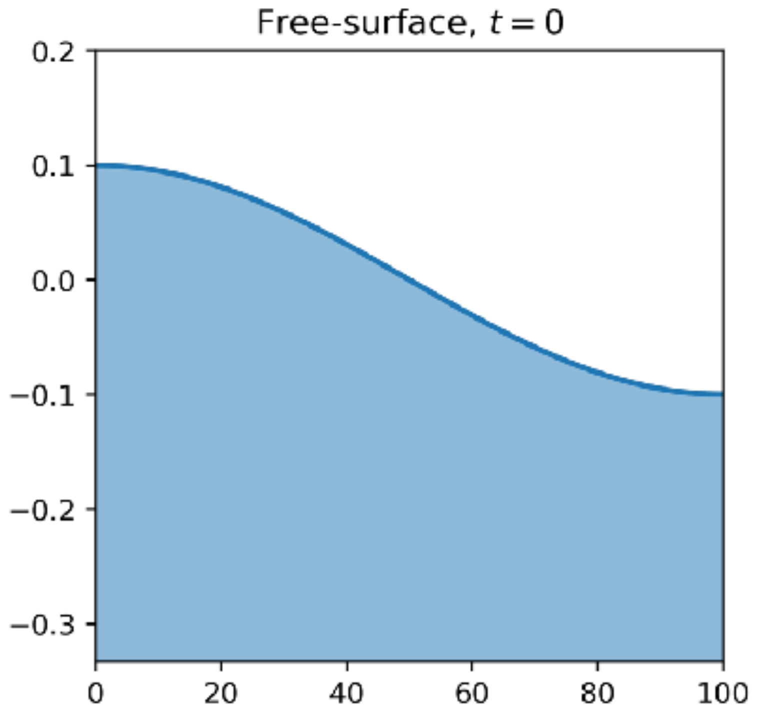
Initial Condition, Parameters:

$$\eta(t=0) = a\cos(kx)$$

$H = 10$ m $L = 100$ m

$k = \dfrac{\pi}{L}$ $a = 0.1$ m

$T = 1$ s $\omega = \dfrac{2\pi}{T}$



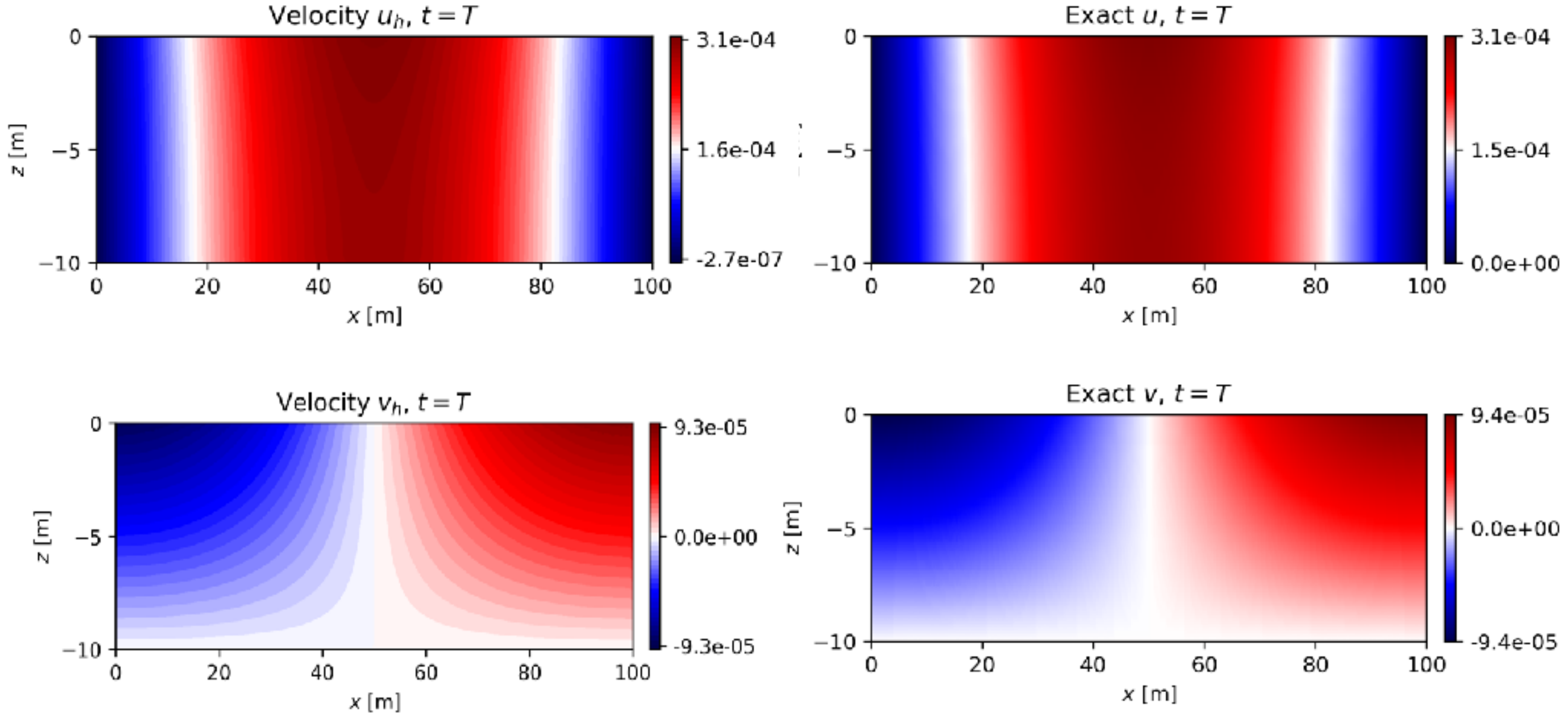
Analytical Solution: (linear gravity-wave theory)

$$u = ag\frac{k}{\omega}\frac{\cosh k(z+H)}{\cosh(kH)}\sin(kx)\sin(\omega t)$$

$$v = -ag\frac{k}{\omega}\frac{\sinh k(z+H)}{\cosh(kH)}\cos(kx)\sin(\omega t)$$

Solutions at one seiche period (t=T)

Numerical **Exact**

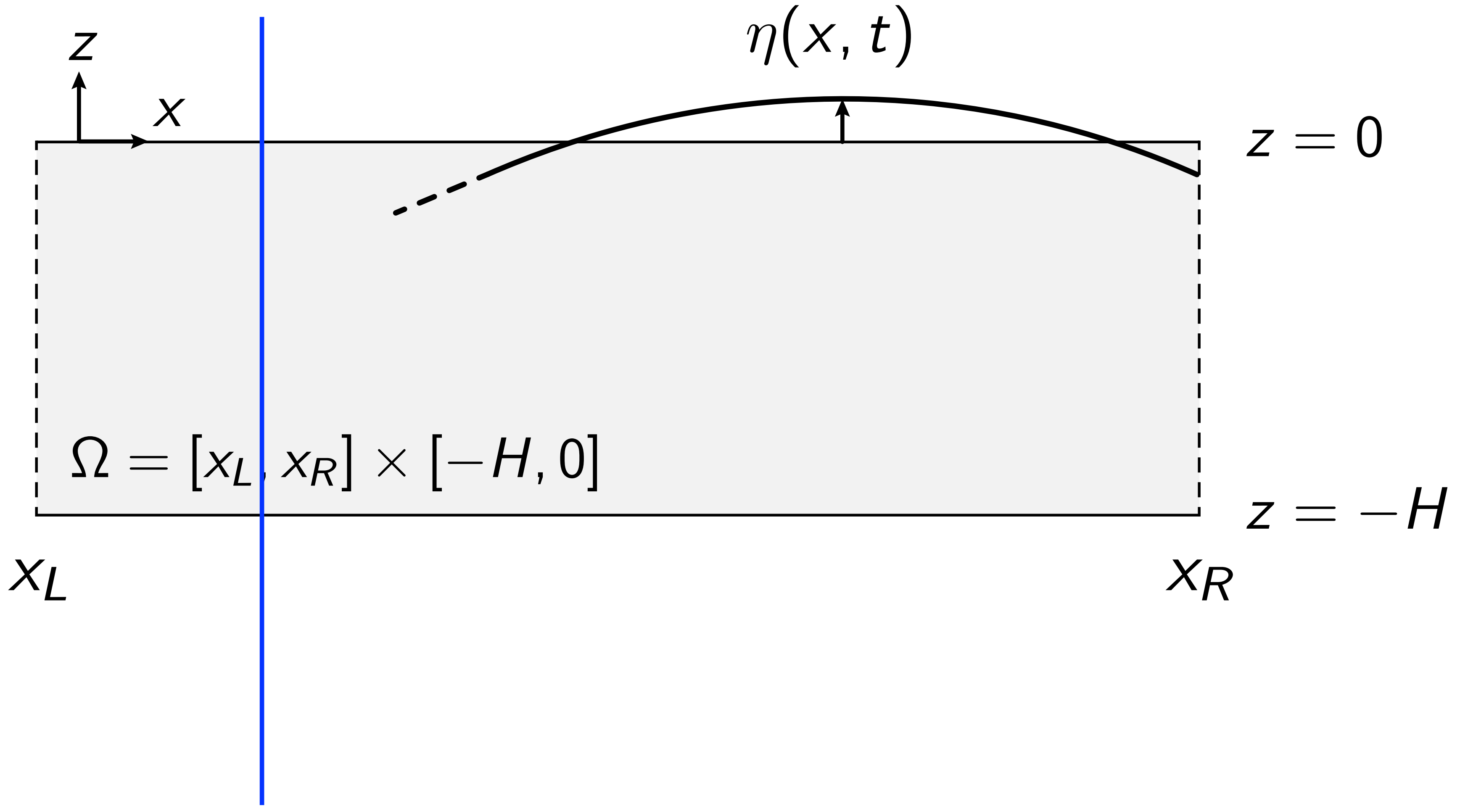


Verification of NHS model: free-surface seiche

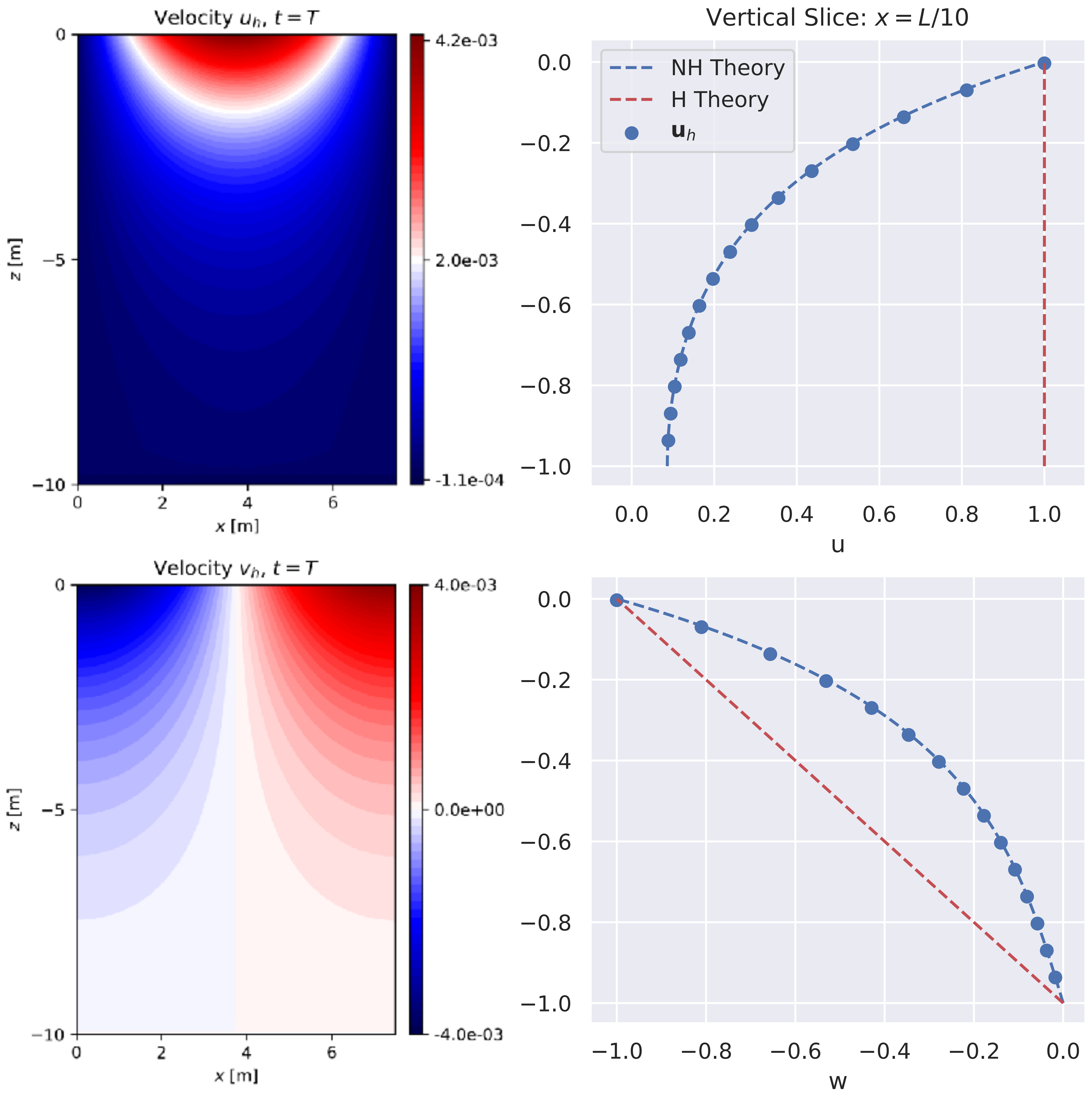
Initial condition, parameters:

$$\eta(t=0) = a\cos(kx)$$

$H = 10$ m $\quad L = 7.5$ m $\quad a = 0.1$ m $\quad T = 10^{-2}$ s



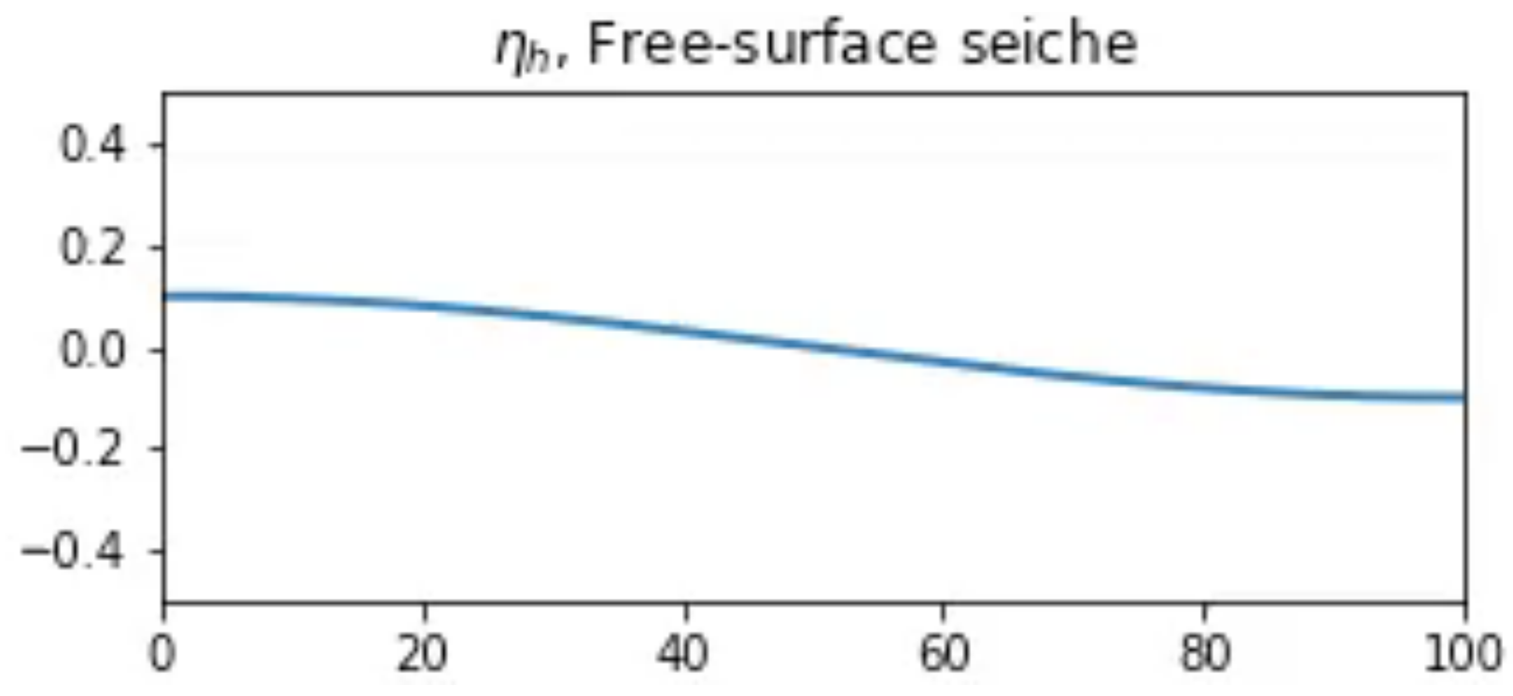
vertical slice: x=L/10

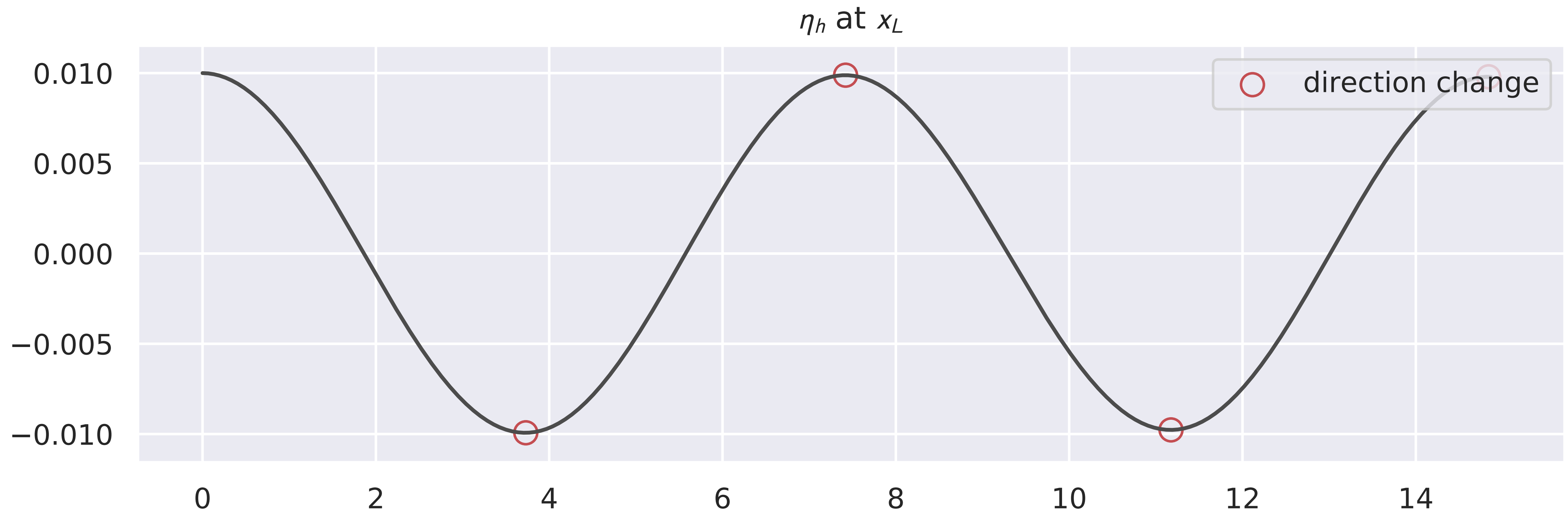


Verification of NHS model: free-surface seiche

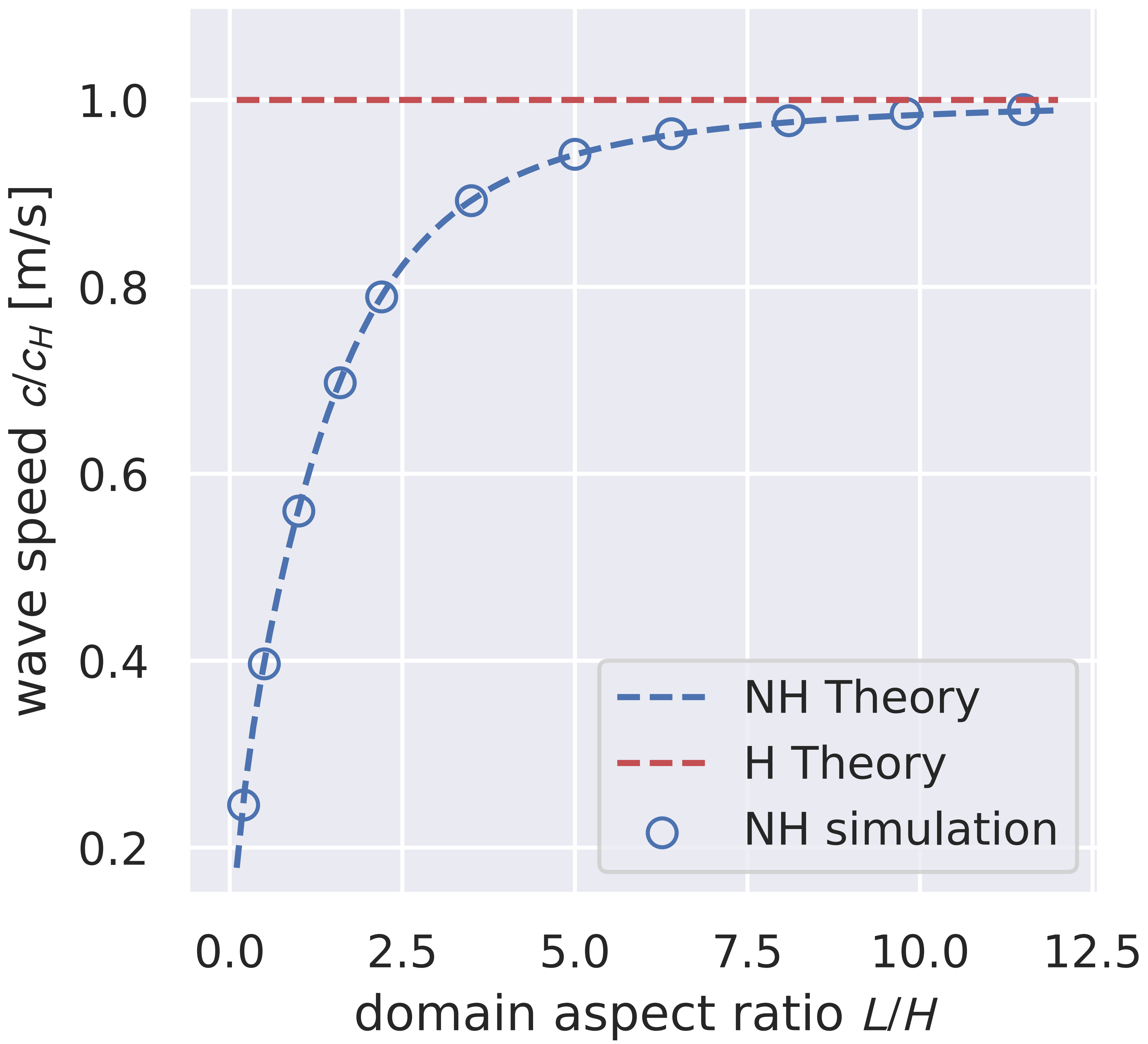
Depth profiles: agreement with NHS theory, deviation from HS prediction

Free-surface: numerical wave speed c





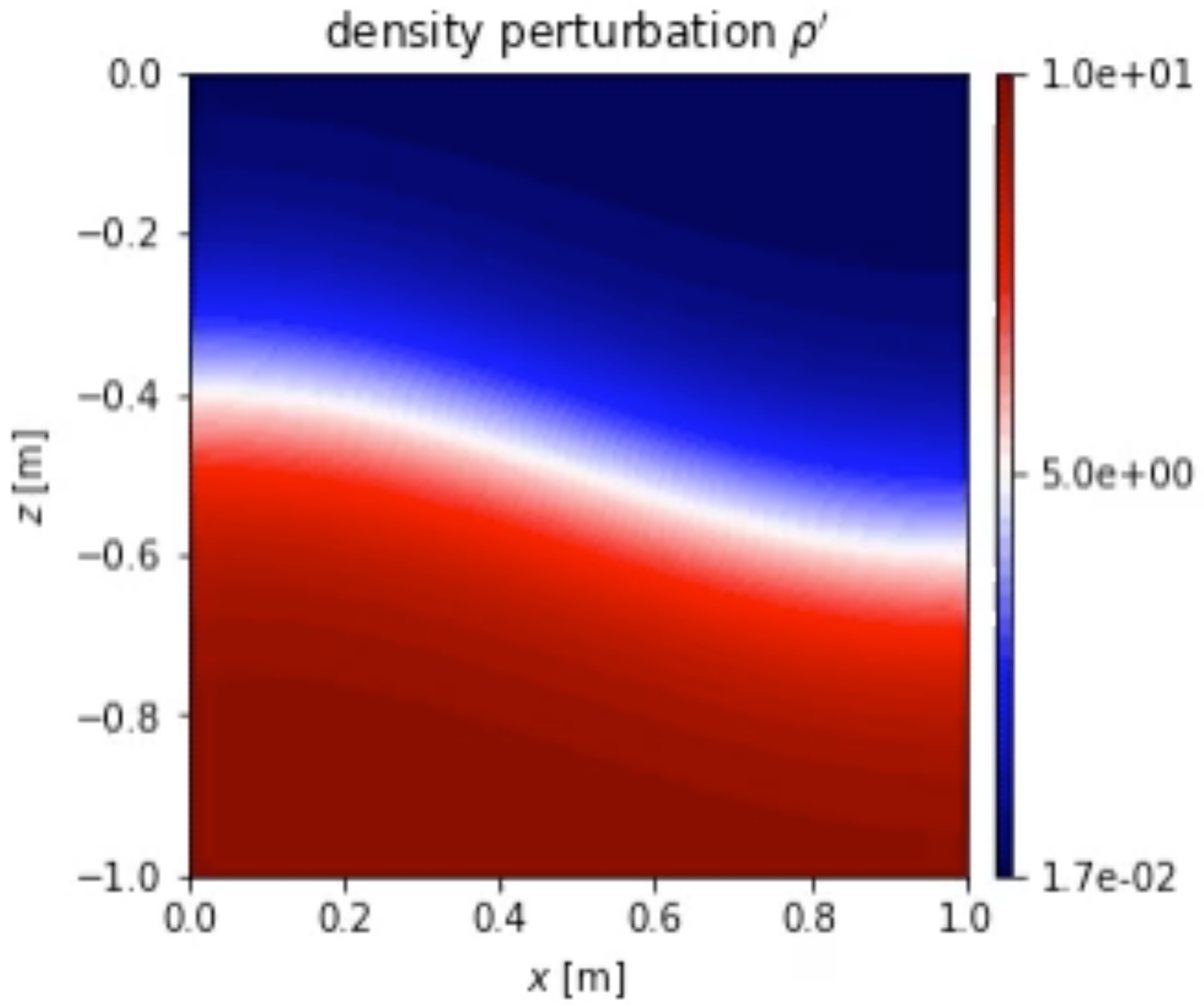
$$c = \sqrt{\frac{g}{k}\tanh(kH)}$$

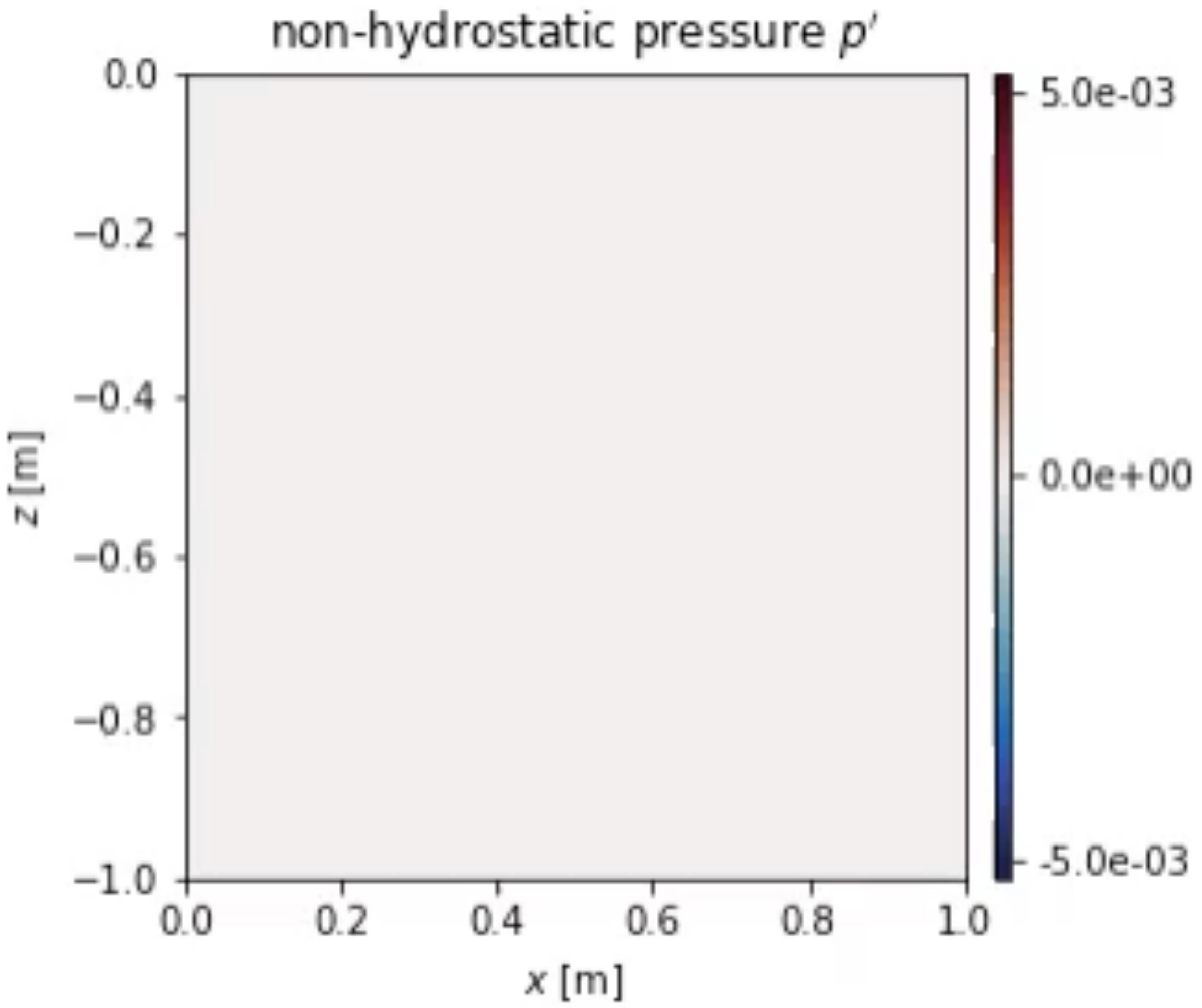


Continuously-stratified internal seiche

Initial Condition:

Zero-free surface, stratified density perturbation $$\rho' = \frac{\Delta\rho}{2}\left[1 - \tanh\left(\frac{2\tanh^{-1}\alpha_s}{\delta_\rho}(z + H/2 - \xi)\right)\right]$$





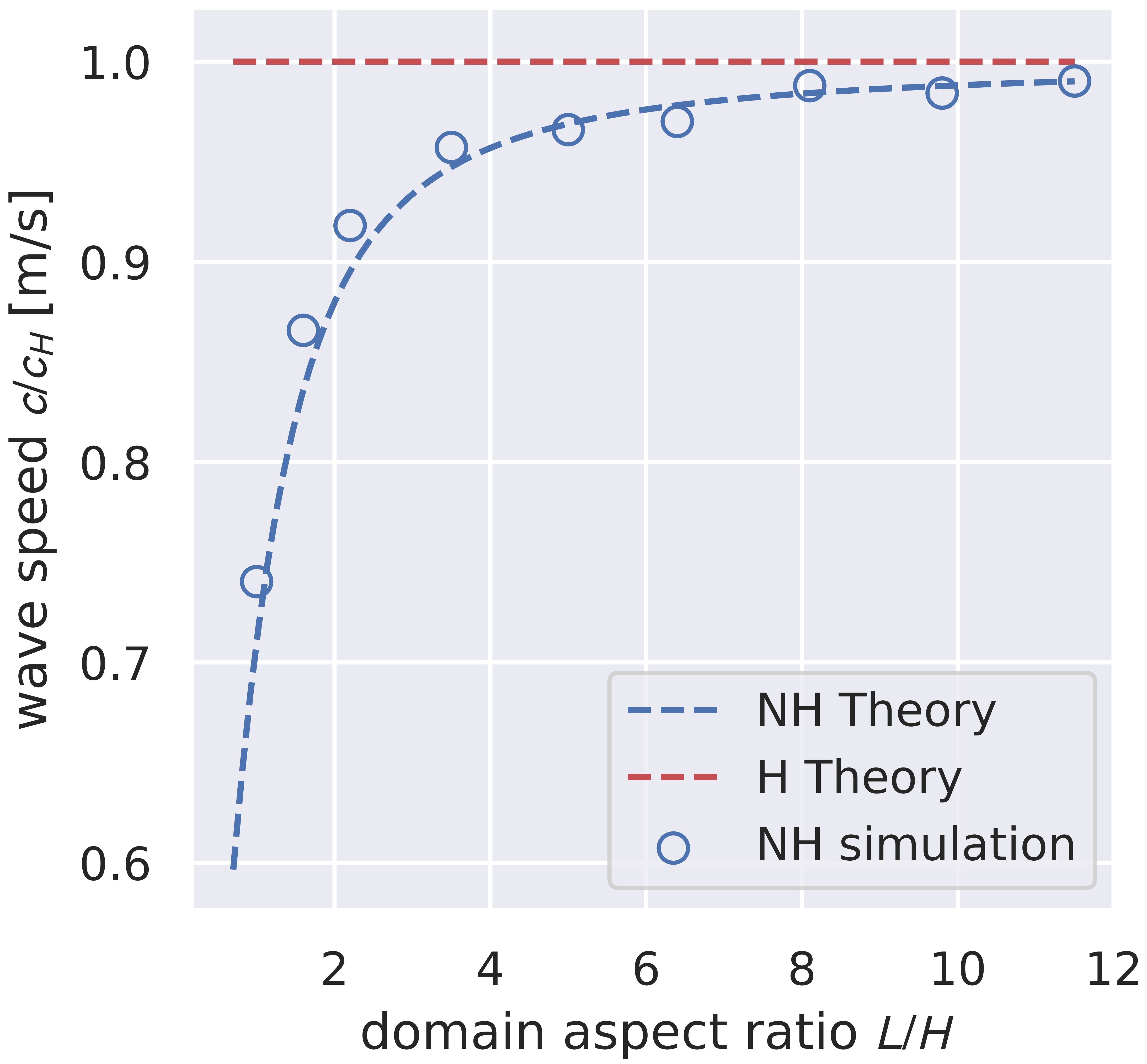
[1] S. Vitousek and O.B. Fringer (2014), OM [2] **Foucart** et al., IEEE Oceans 2021 [3] **Foucart** et al., Ocean Modelling. in prep. 2023a.

Continuously-stratified internal seiche

Interface wave speeds agree with theoretical values

$$c = \sqrt{\frac{g'}{2k}\tanh\left(\frac{kH}{2}\right) f_i(k\delta\rho)}$$

...over many different simulations with differing domain aspect ratios.

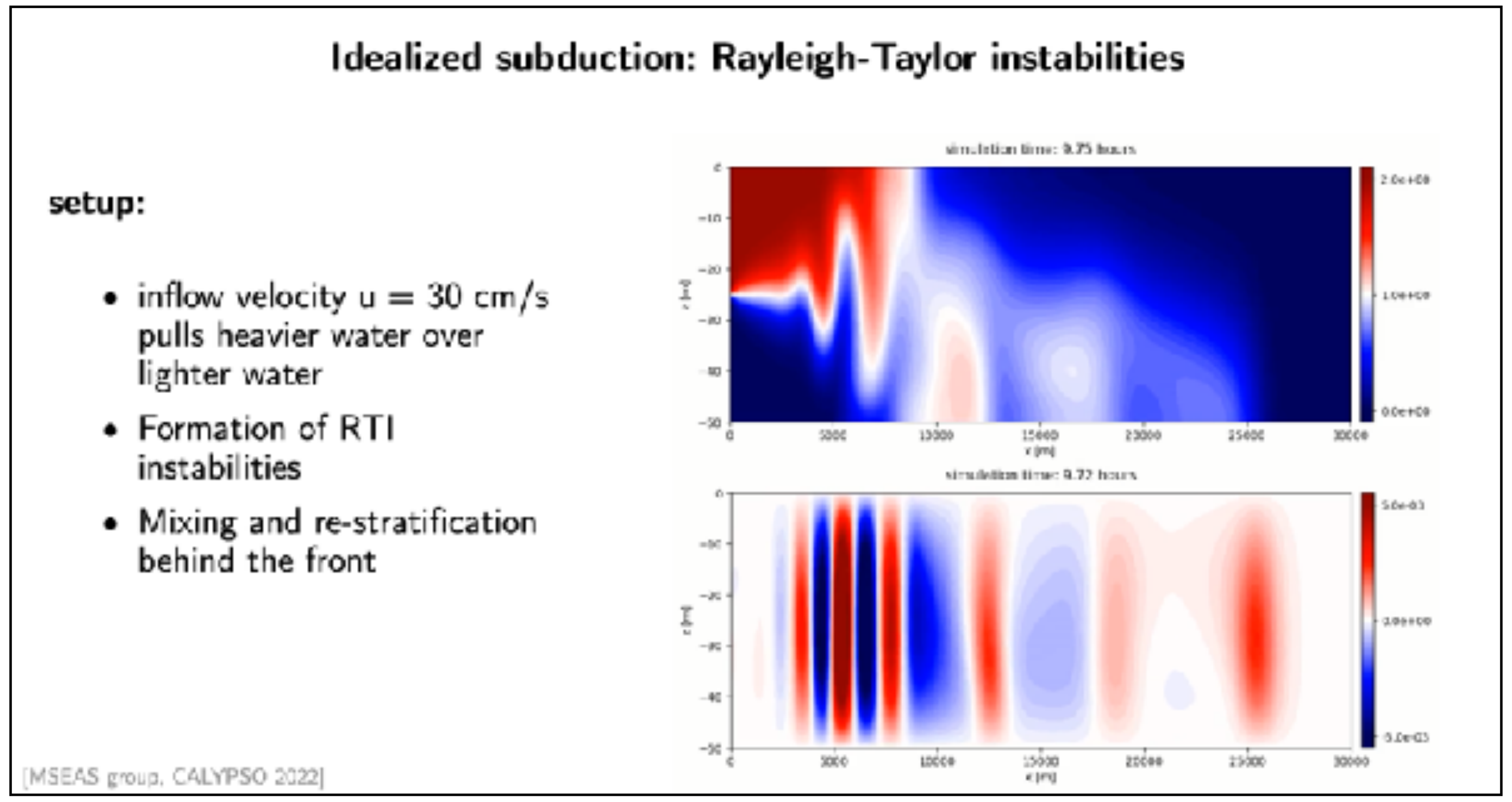


[1] S. Vitousek and O.B. Fringer (2014), OM [2] **Foucart** et al., IEEE Oceans 2021 [3] **Foucart** et al., Ocean Modelling. in prep. 2023a.

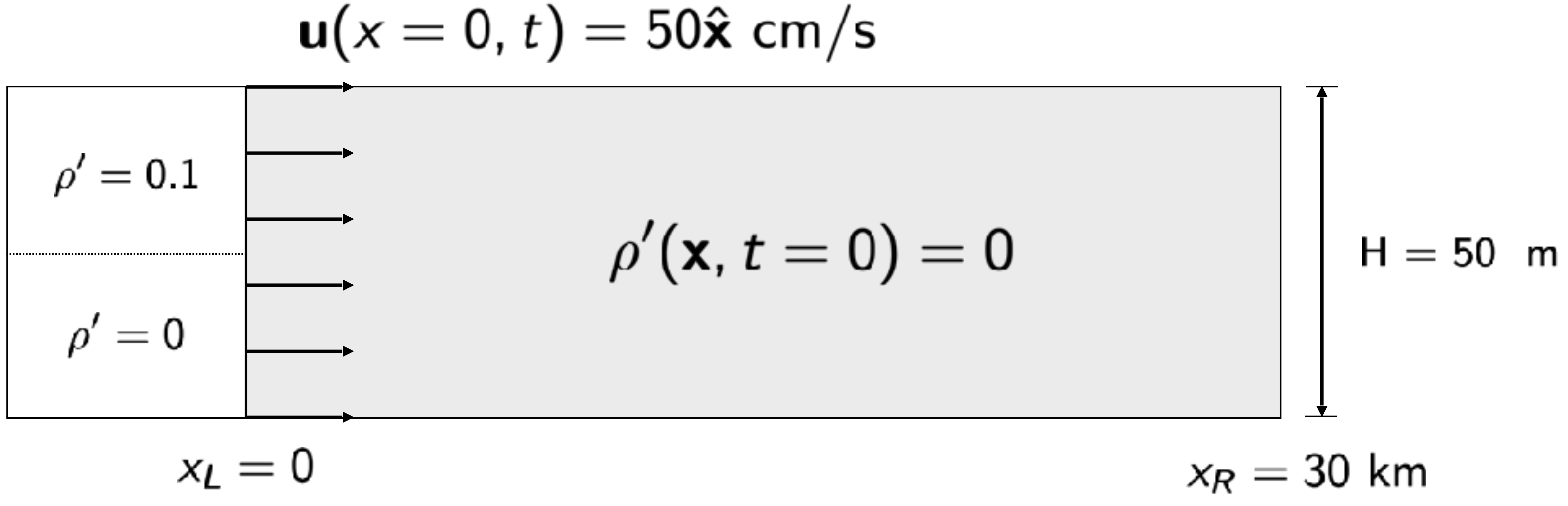
Additional verification / validation of HDG NHS model

Idealized mixed-layer Rayleigh Taylor Instability formation (shown last time)

- inflow velocity u = 50 cm/s pulls heavier water over lighter water
- Formation of RTI instabilities
- Mixing and re-stratification behind the front



Modeling Domain



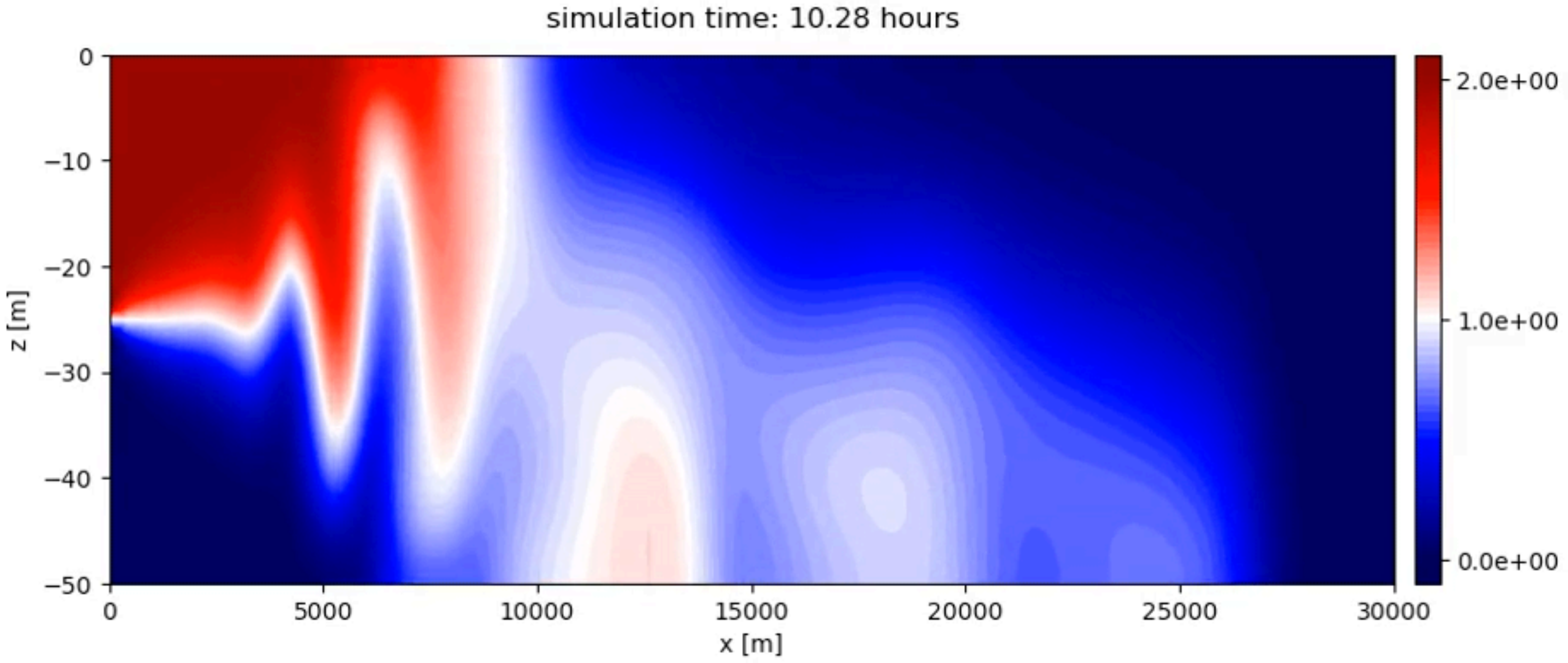
Boundary Conditions

	top	bottom	inflow	ouflow
$\bar{u}$	Γ_N	Γ_N	Γ_D	Γ_N
$\bar{w}$	Γ_N	Γ_D	Γ_D	Γ_N
$\delta\eta$	-	-	Γ_N	Γ_N
$\delta p'$	Γ_D	Γ_N	Γ_N	Γ_N
ρ'	Γ_N	Γ_N	Γ_N	Γ_N

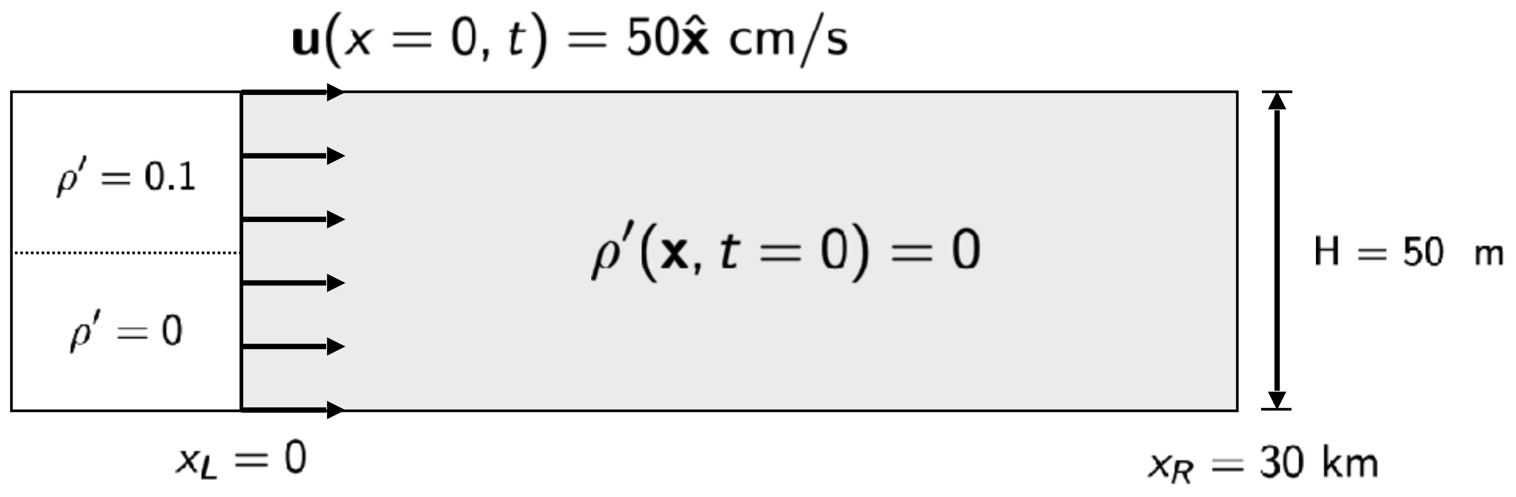
Idealized subduction: Rayleigh-Taylor instabilities

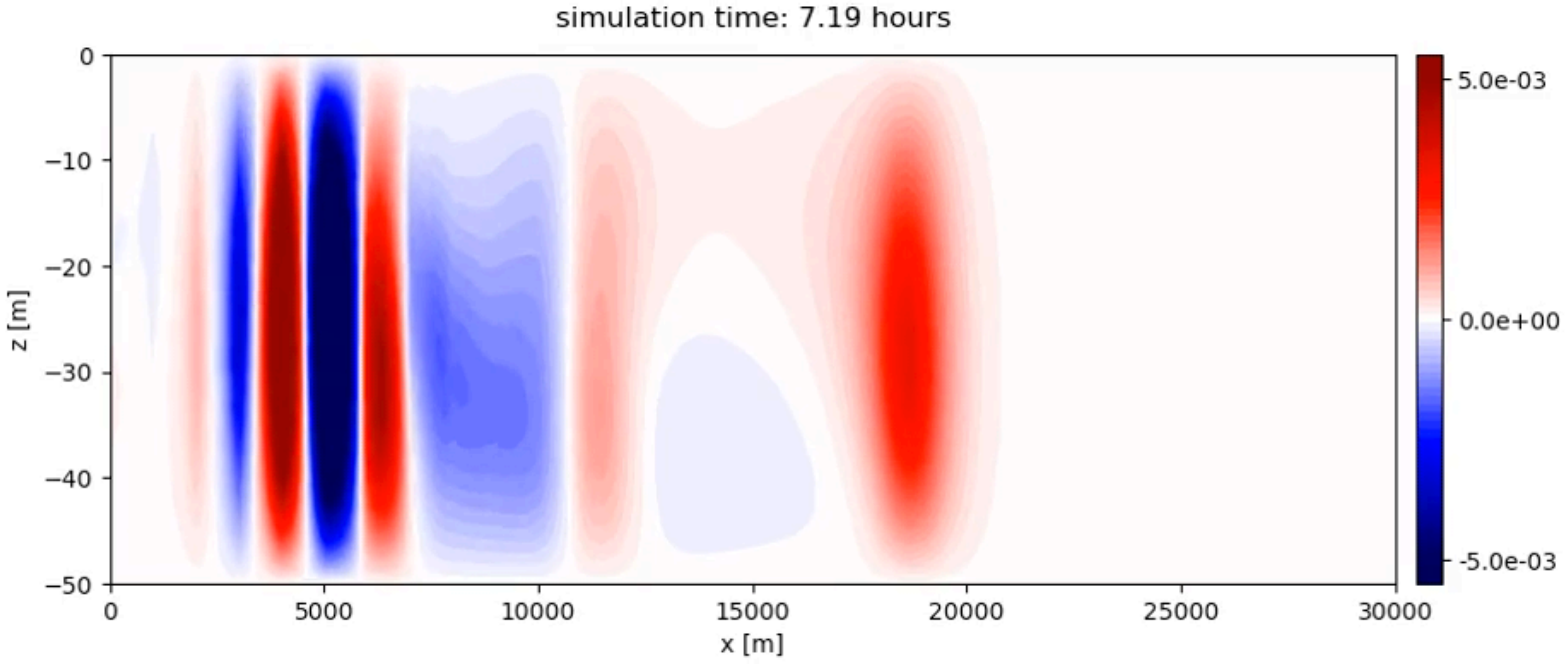
Simulation:

- Formation of RTI instabilities
- Mixing and re-stratification behind the front

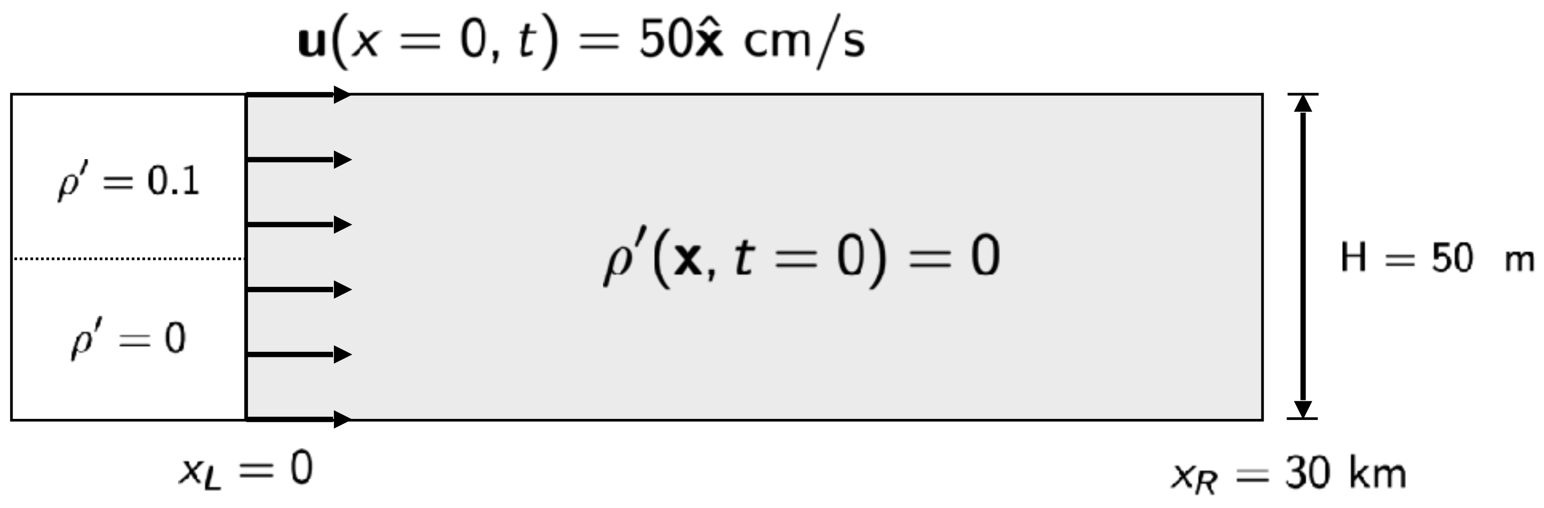


Modeling Domain





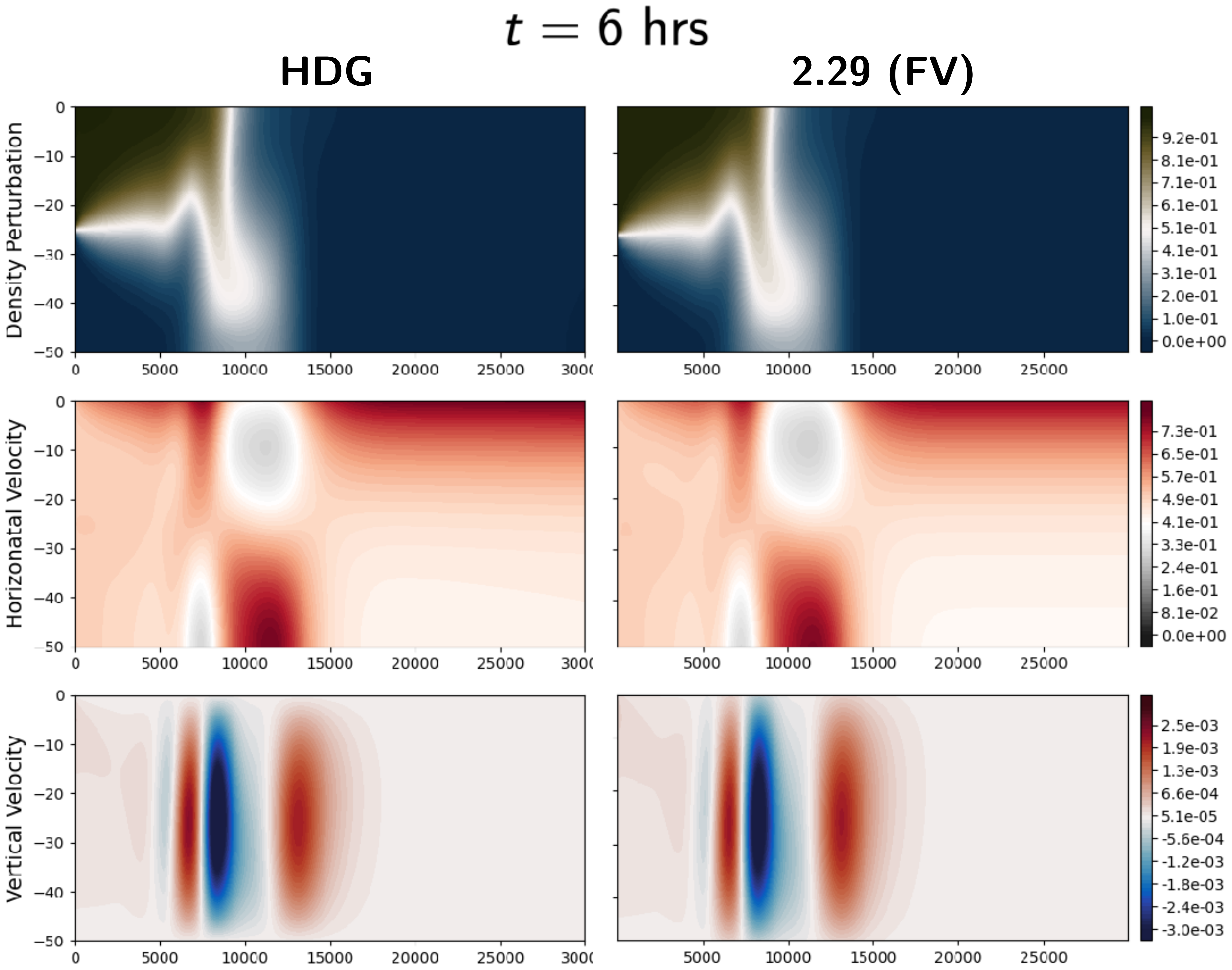
Additional verification / validation of HDG NHS model



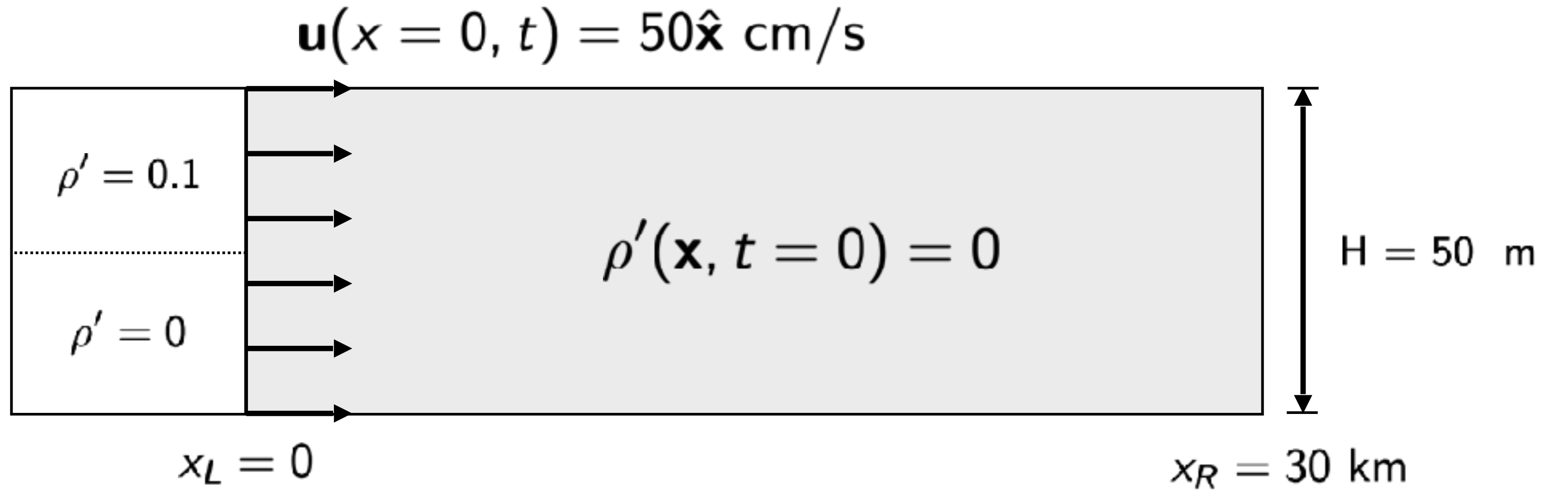
- **Goal:** verify diffusive physics by comparison to a legacy finite volume (FV) code
  - 2D incompressible Navier-Stokes solver with Boussinesq approximation

parameters

	HDG	FV	units
N_x, N_z	$(25, 10)$	$(540, 25)$	
ν_x	$1 \cdot 10^2$	$1 \cdot 10^2$	m^2/s
ν_z	$8 \cdot 10^{-3}$	$8 \cdot 10^{-3}$	m^2/s
Δt	100	100	s
order	4	2	

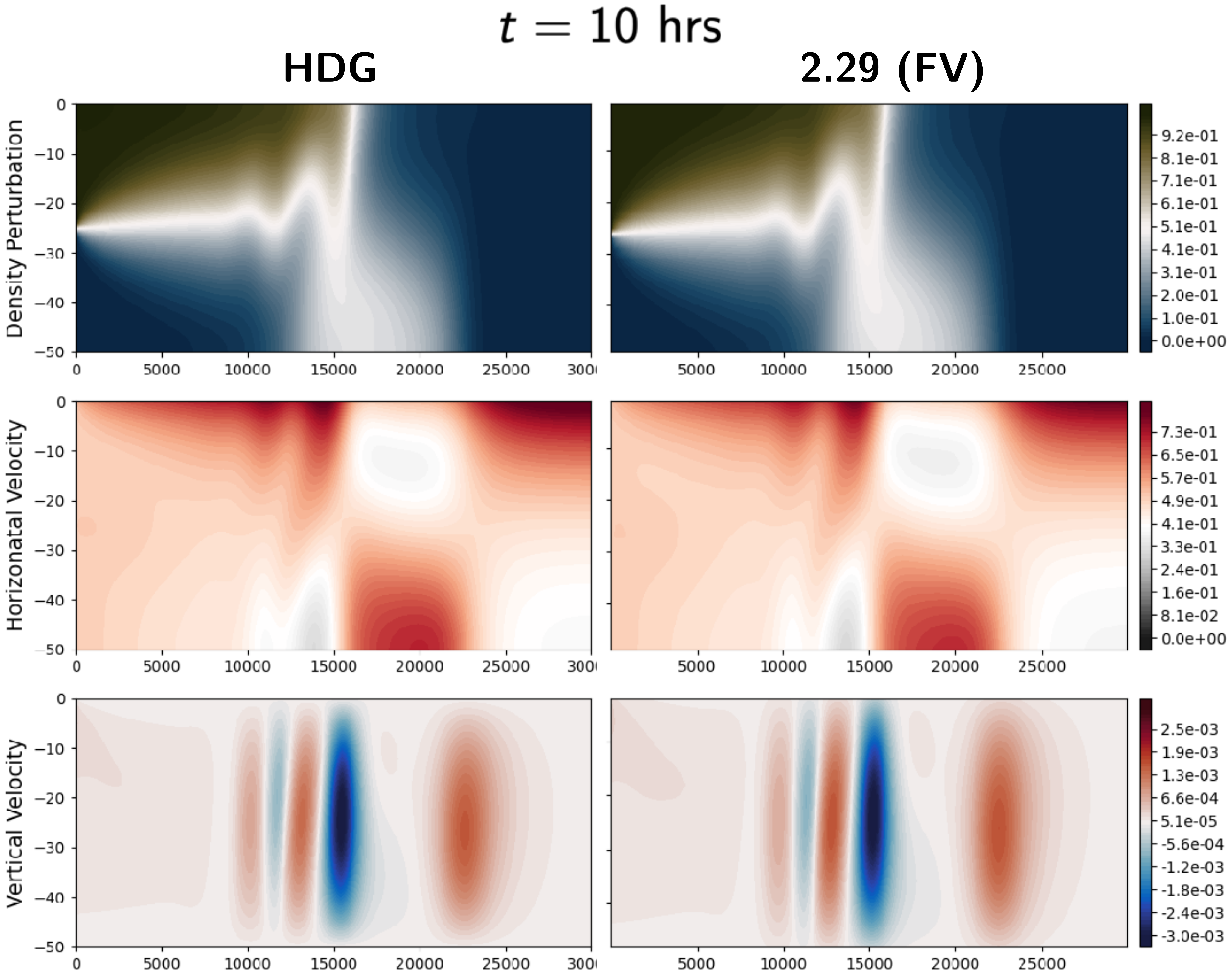


Additional verification / validation of HDG NHS model



- **Goal**: verify diffusive physics by comparison to a legacy finite volume (FV) code
  - 2D incompressible Navier-Stokes solver with Boussinesq approximation

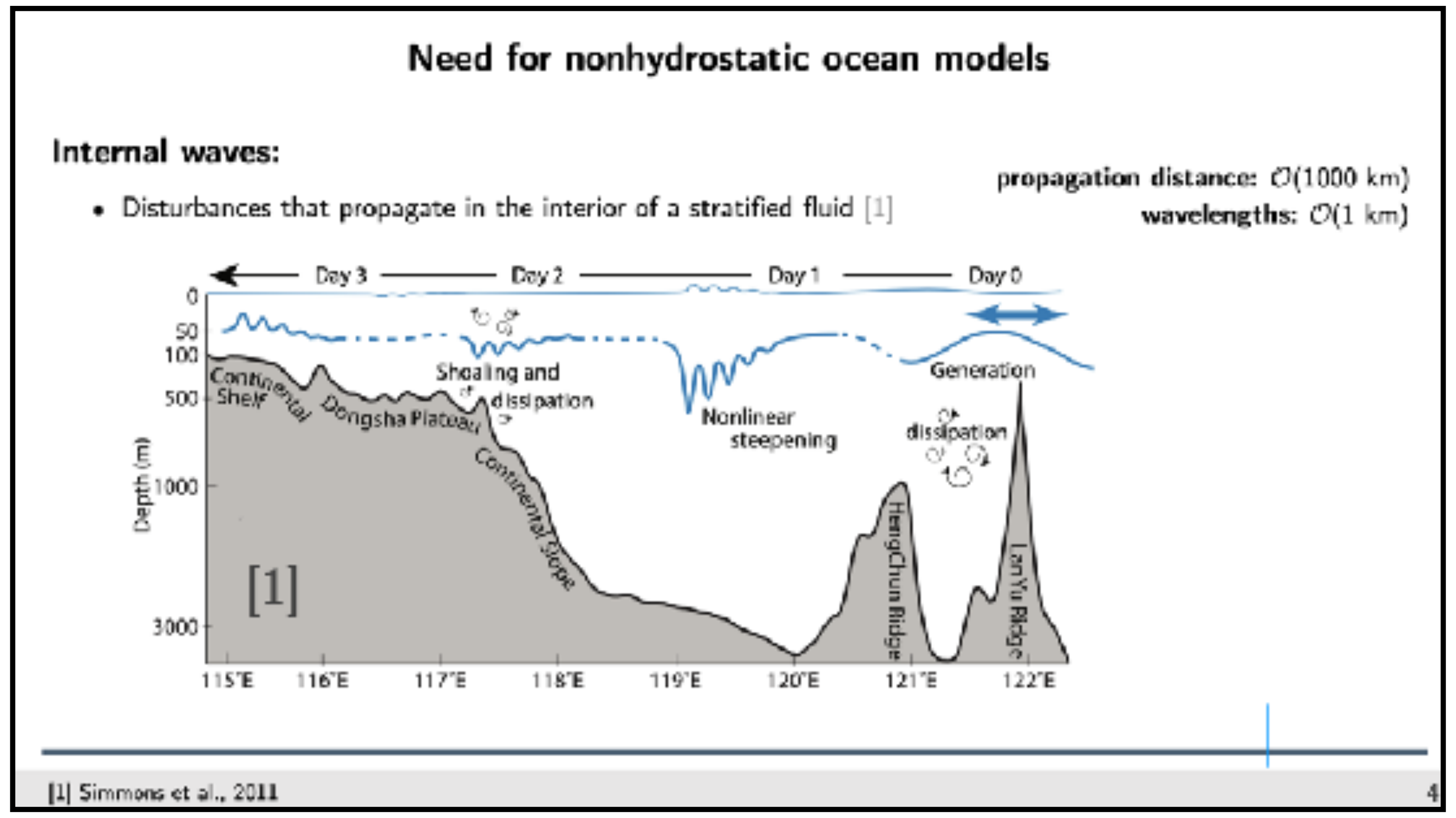
- Differences explainable by free surface
- Excellent agreement with FV code
- HDG: significant reduction of cost (high-order)
  - 540 x 25 grid (FV)
  - 25 x 10 elements (polynomial order 4)



[1] **Foucart** et al., Ocean Modelling. in prep. 2023a.

Internal Solitary Wave Generation

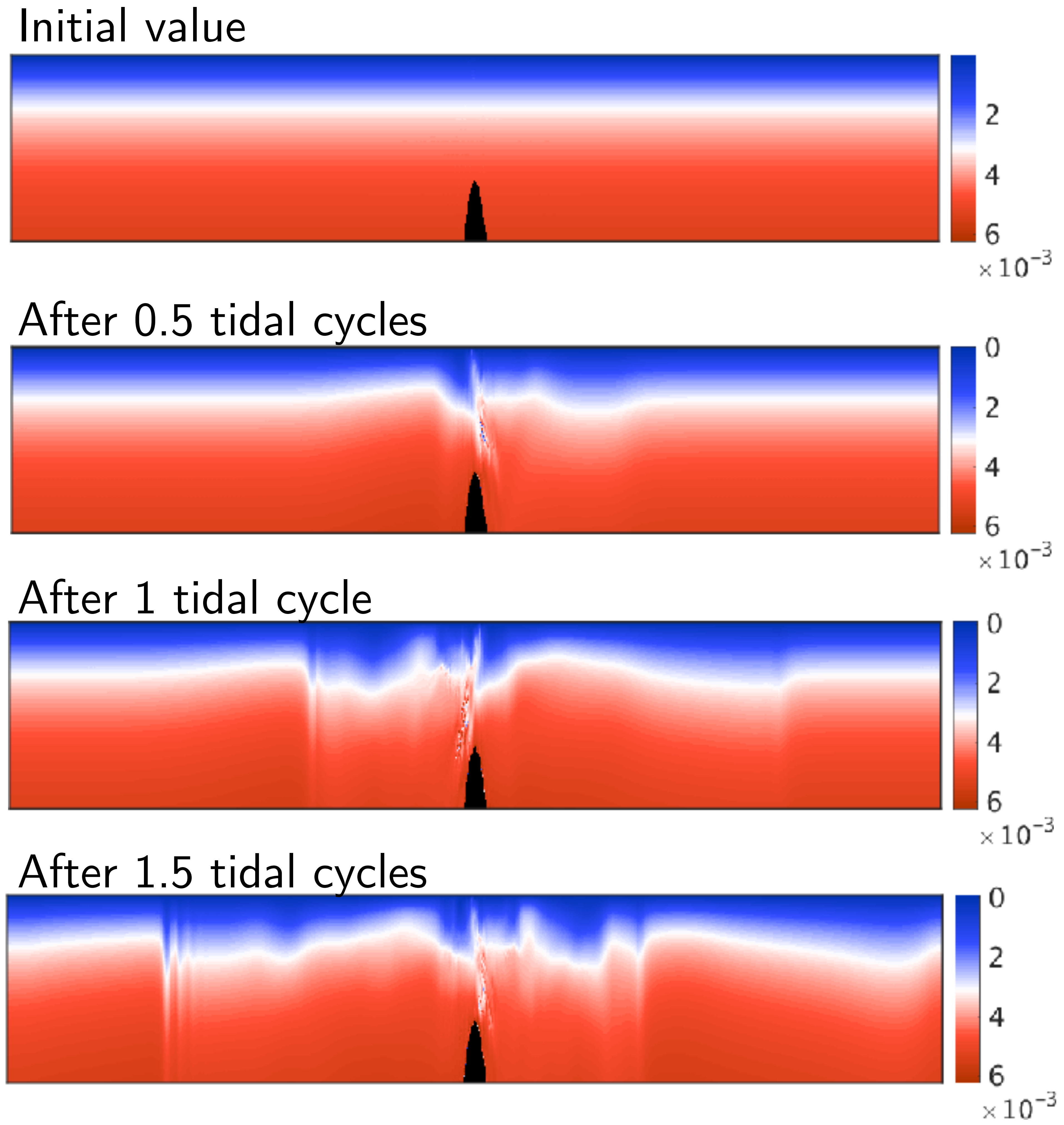
Strongly-stratified flow over "tall" seamount:



- 200km x 1000m domain
- ISWs propagate away from the seamount
- Nonlinear ISW train develops in leading left-propagating wave starting around 1.25 tidal cycles
- Qualitative agreement with expected nonhydrostatic behavior

Low order simulations **did not** capture ISW train

Normalized density perturbation ρ'/ρ_0
(Top 20% of domain)



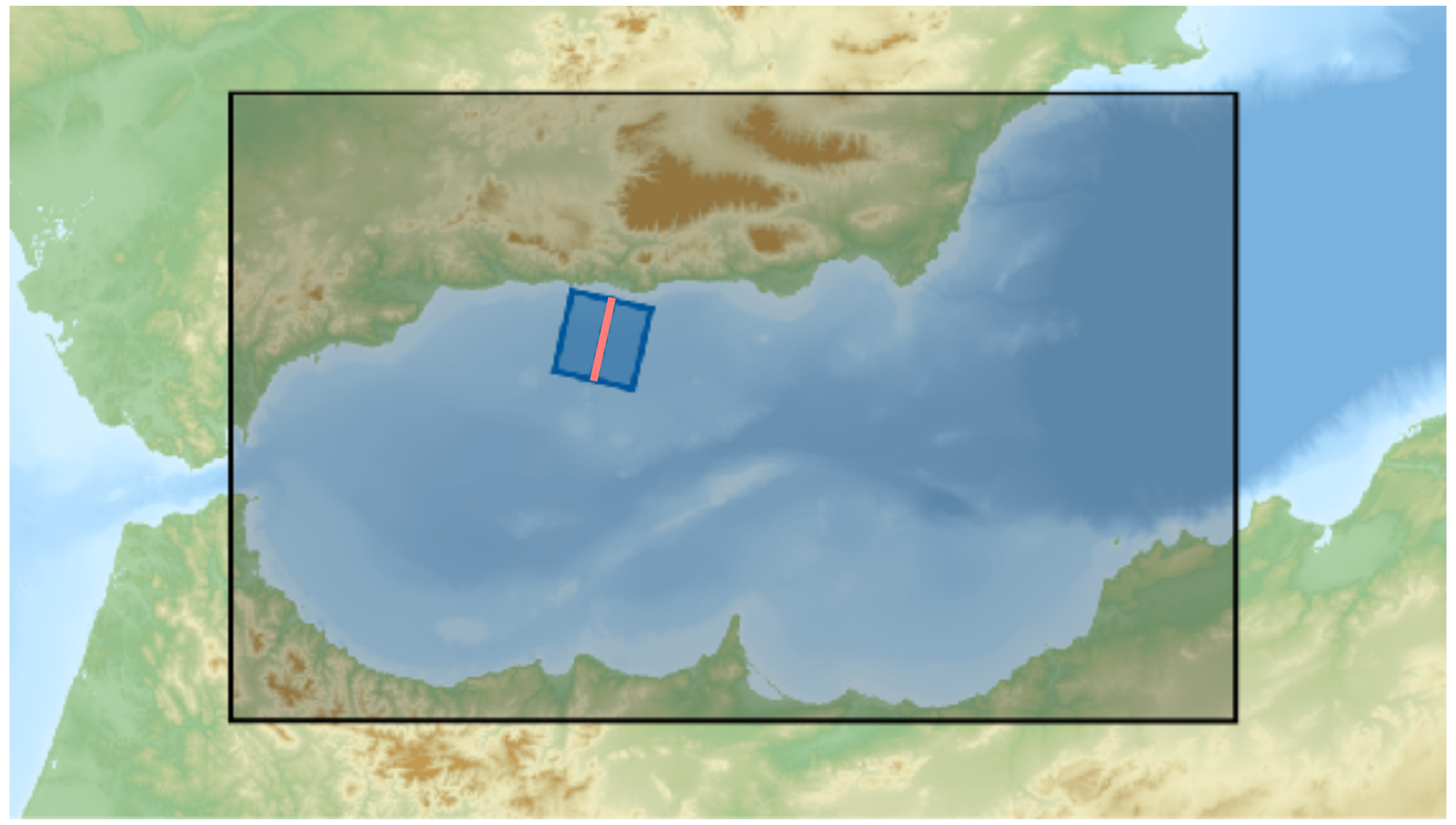
Internal Solitary Wave Generation

Application: 2D model nesting within a large hydrostatic code (MSEAS-PE)

Region of investigation:

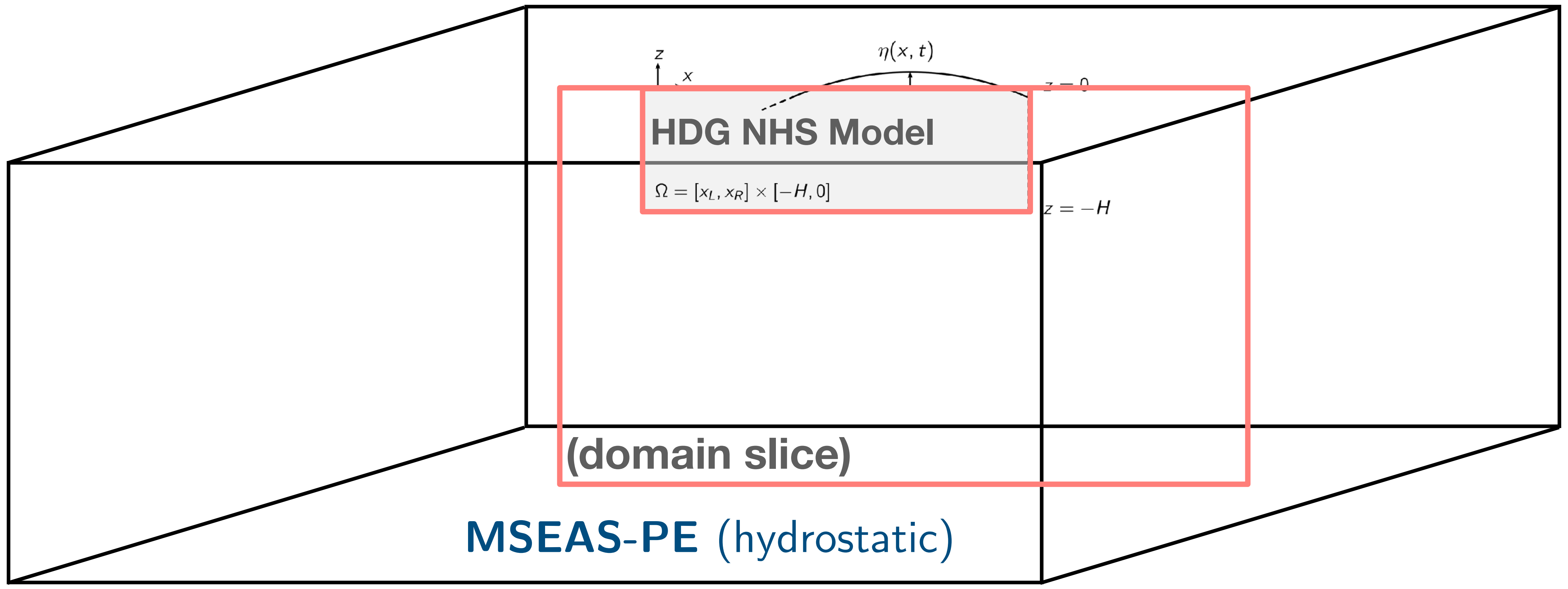
- **Salmon-colored line:** denotes a vertical section in the Alboran Sea along which **wind-driven instabilities were observed** in the hydrostatic MSEAS PE simulations
- The section starts in the west along the northern edge of the West Alboran Gyre and extends to the east-by-northeast out of the gyre towards the Spanish coast.
- Implemented 2D HDG NHS model nesting and initialization from real data
- Goal: model instabilities in the mixed layer and compare to HS simulation output

Modeling domains



Modeling domain for model nested runs in the Alboran Sea showing the MSEAS-PE domain (shaded, black), and the HDG NHS nested modeling domain (blue) in 2D (center slice) and 3D (box).

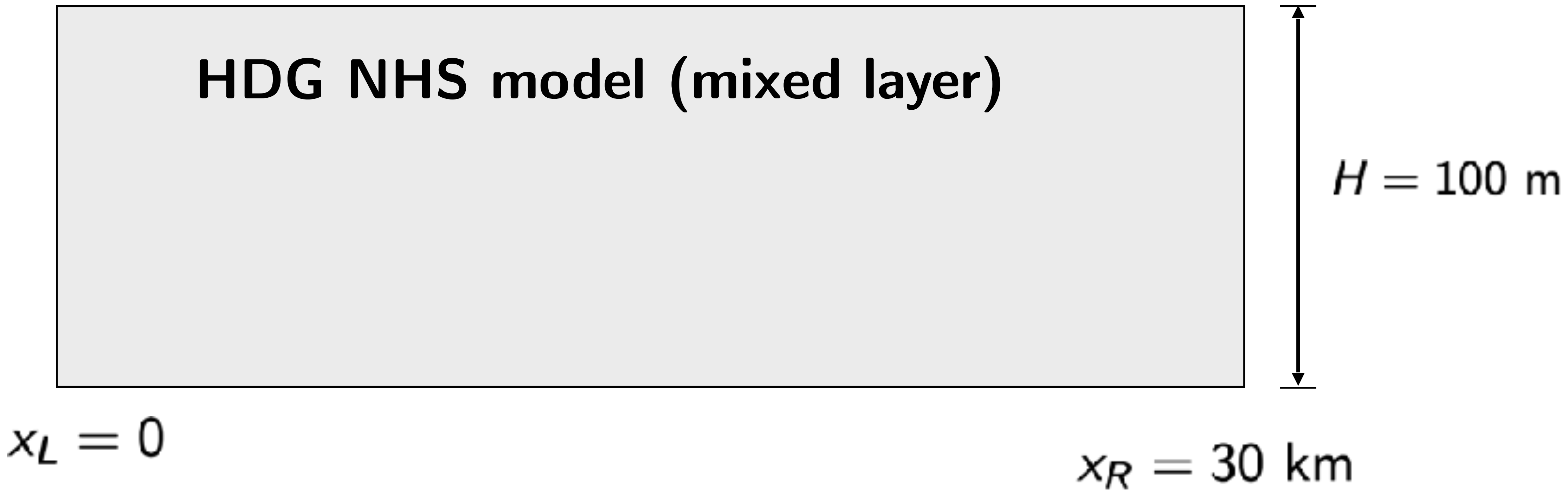
(not to scale)



2D model nesting in the Alboran sea

Model comparison after wind event

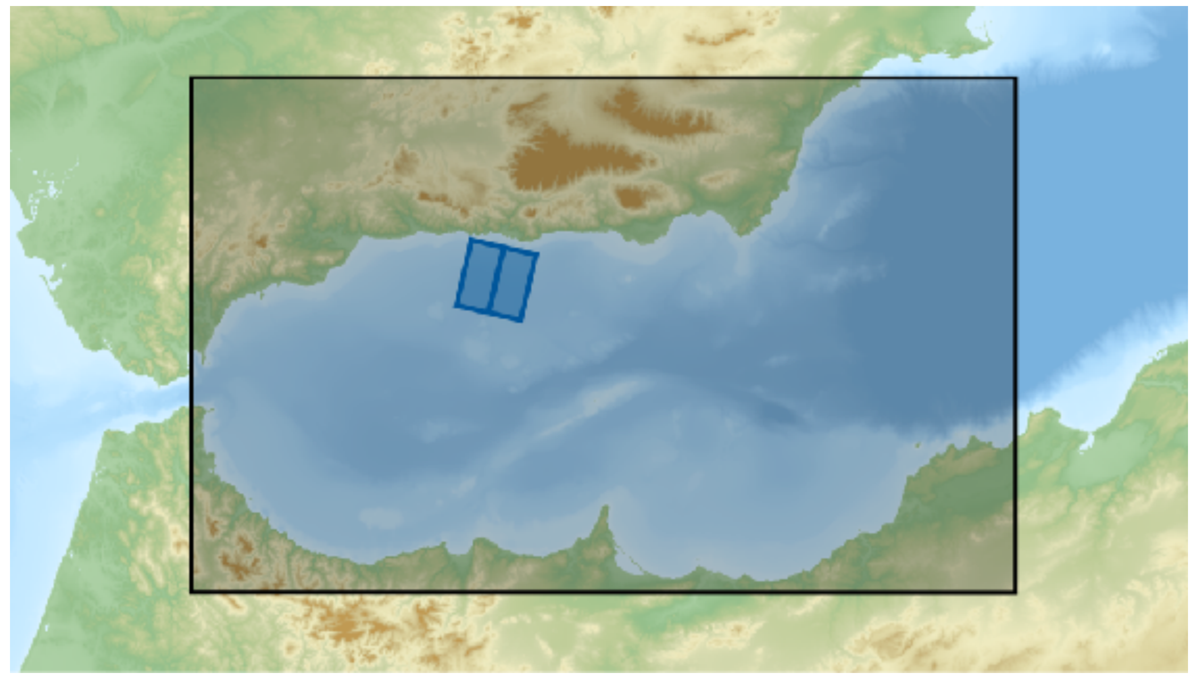
- HDG NHS initialized on March 25, run for 5 days
- Same parameters and effective resolution as hydrostatic MEAS-PE model (500 m)
- Gale: March 27 2019



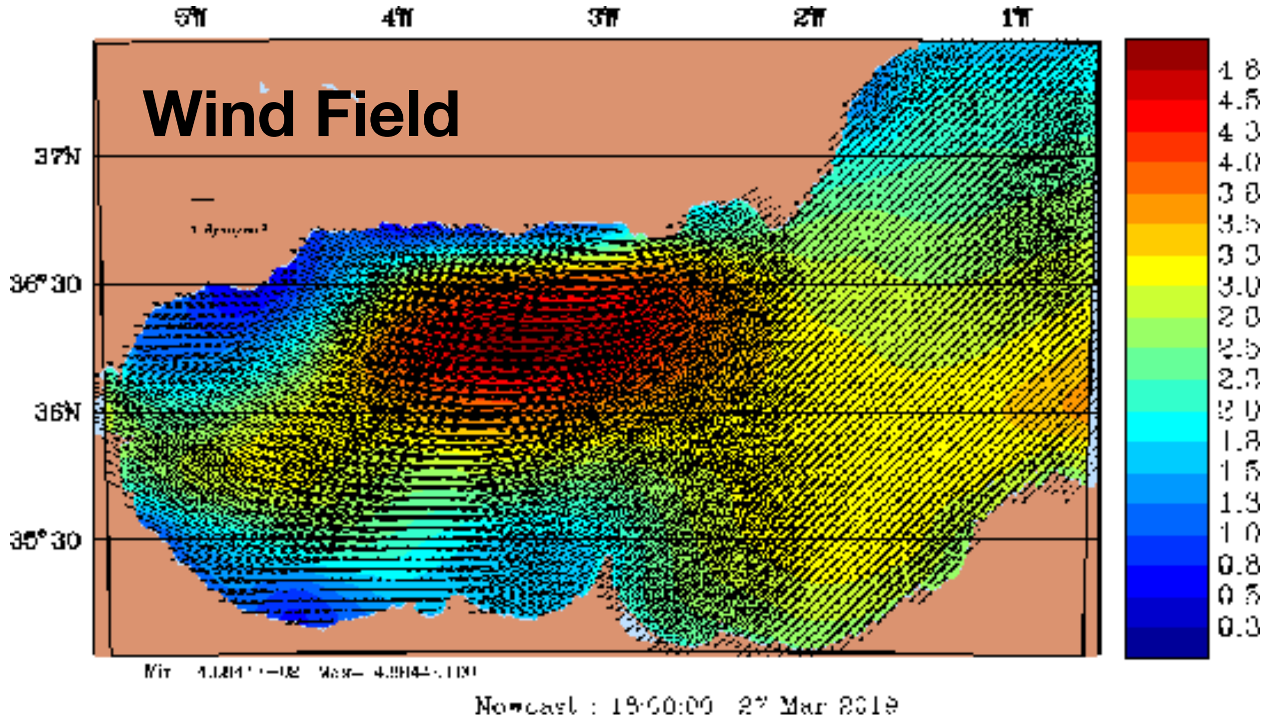
NHS model: boundary conditions

	top	bottom	sides
$\bar{u}$	Γ_D	Γ_D	Γ_D
$\bar{w}$	Γ_N	Γ_D	Γ_D
$\delta\eta$	-	-	Γ_N
$\delta p'$	Γ_D	Γ_N	Γ_N
ρ'	Γ_D	Γ_D	Γ_D

Modeling Domain



March 27, 2019

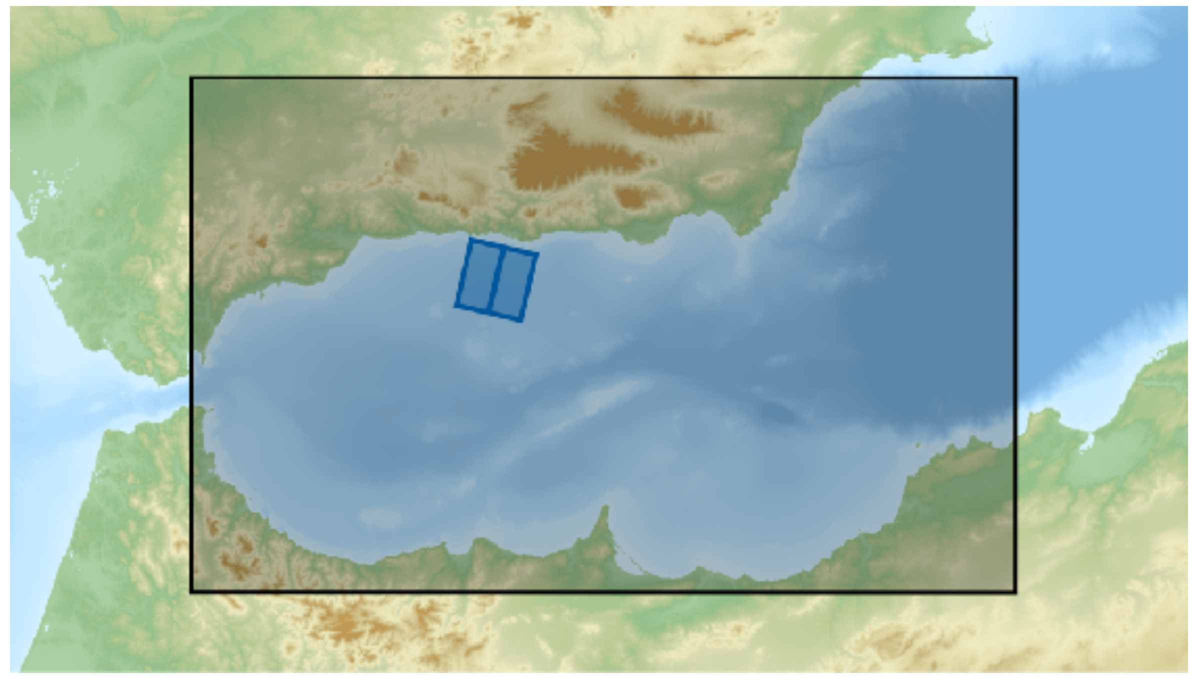


2D model nesting in the Alboran sea

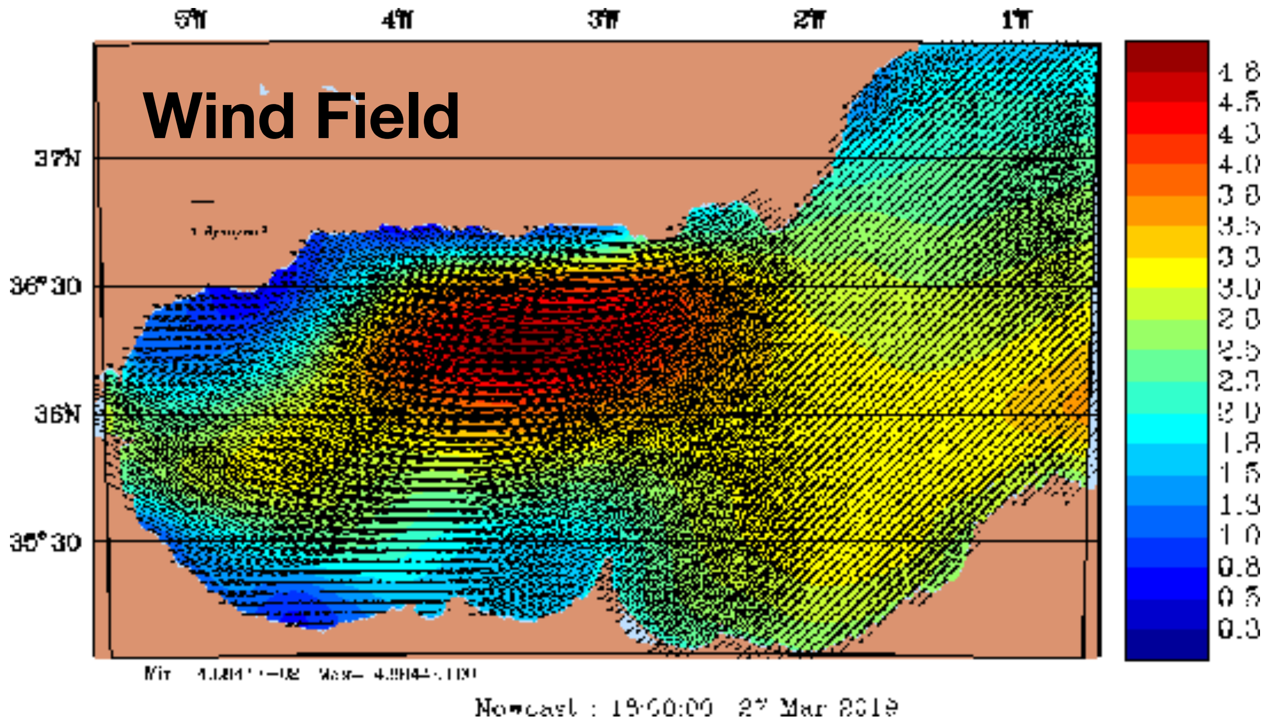
Model comparison after wind event

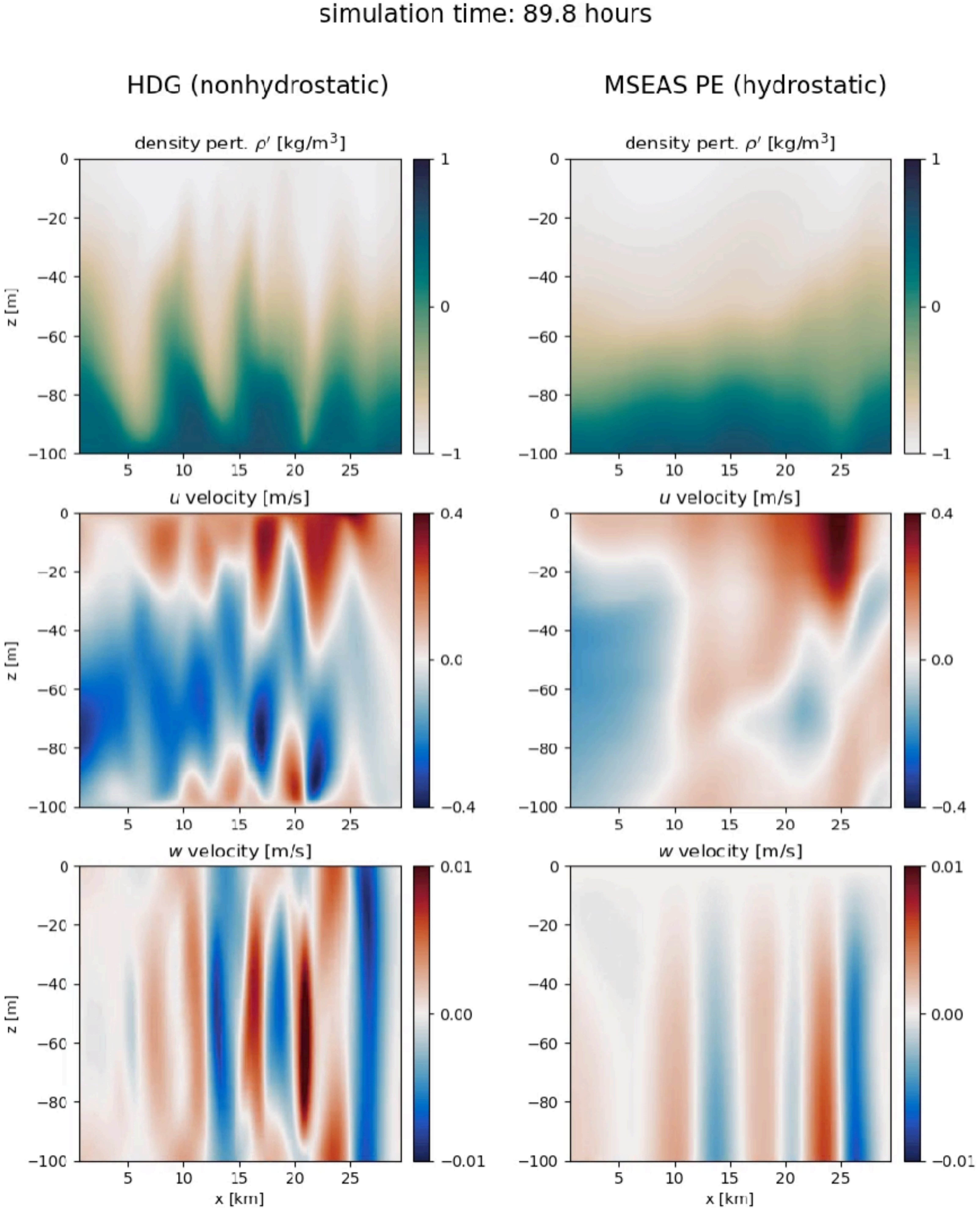
- HDG NHS initialized on March 25, run for 5 days
- Same parameters and effective resolution as hydrostatic MEAS-PE model
- Gale: March 27
- Strong nonhydrostatic response observed over model domain
- After conclusion of wind event, agreement with hydrostatic model once again

Modeling Domain



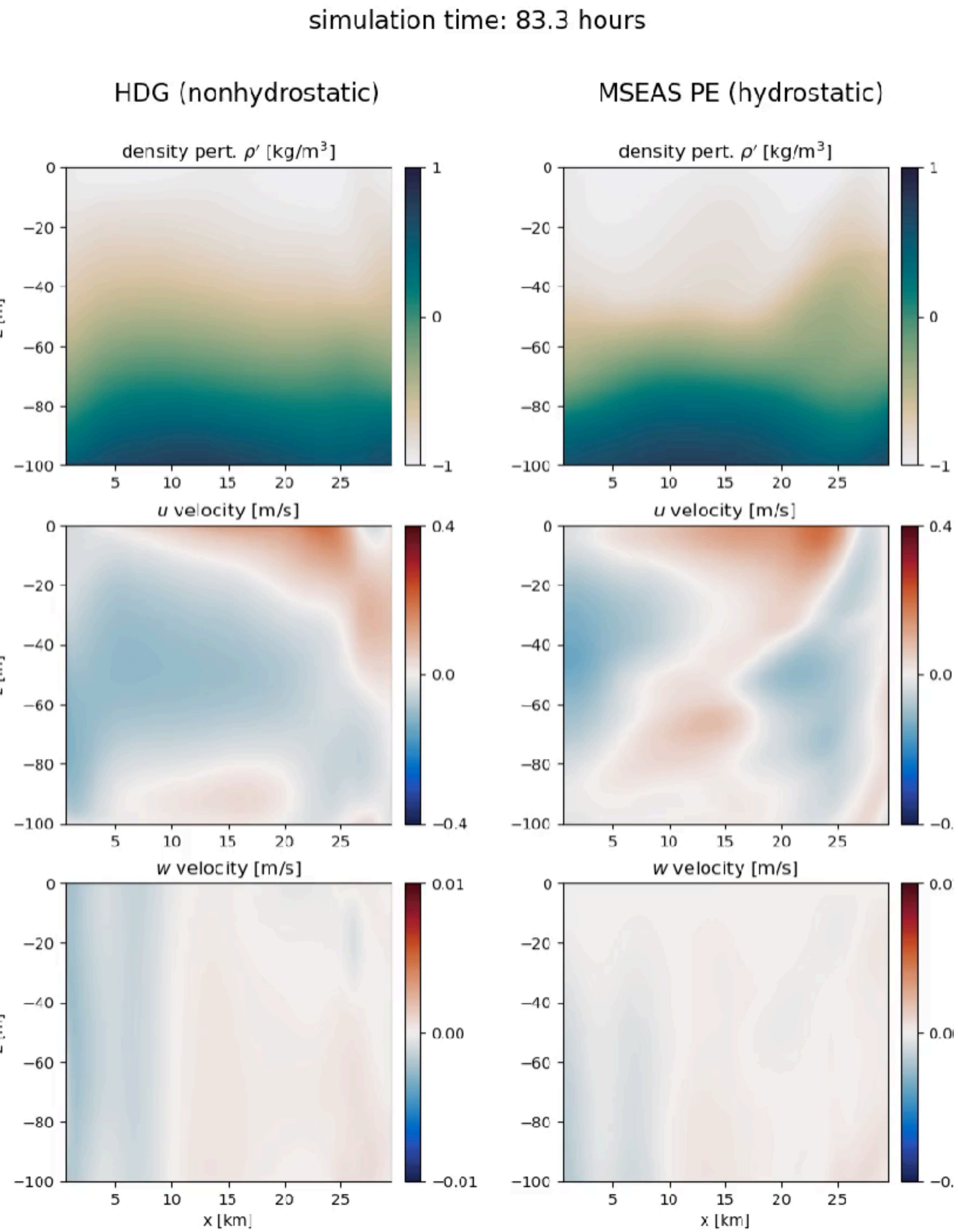
March 27, 2019

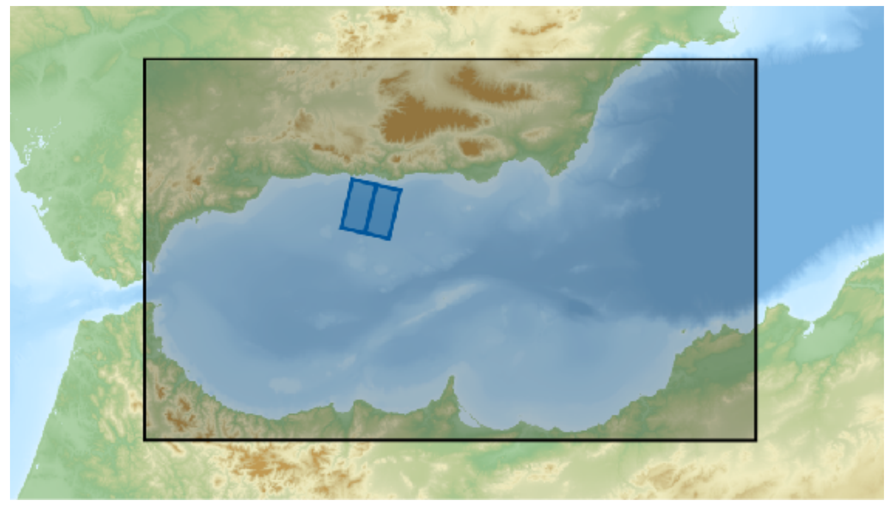




[1] **Foucart** et al., Ocean Modelling. in prep. 2023a.

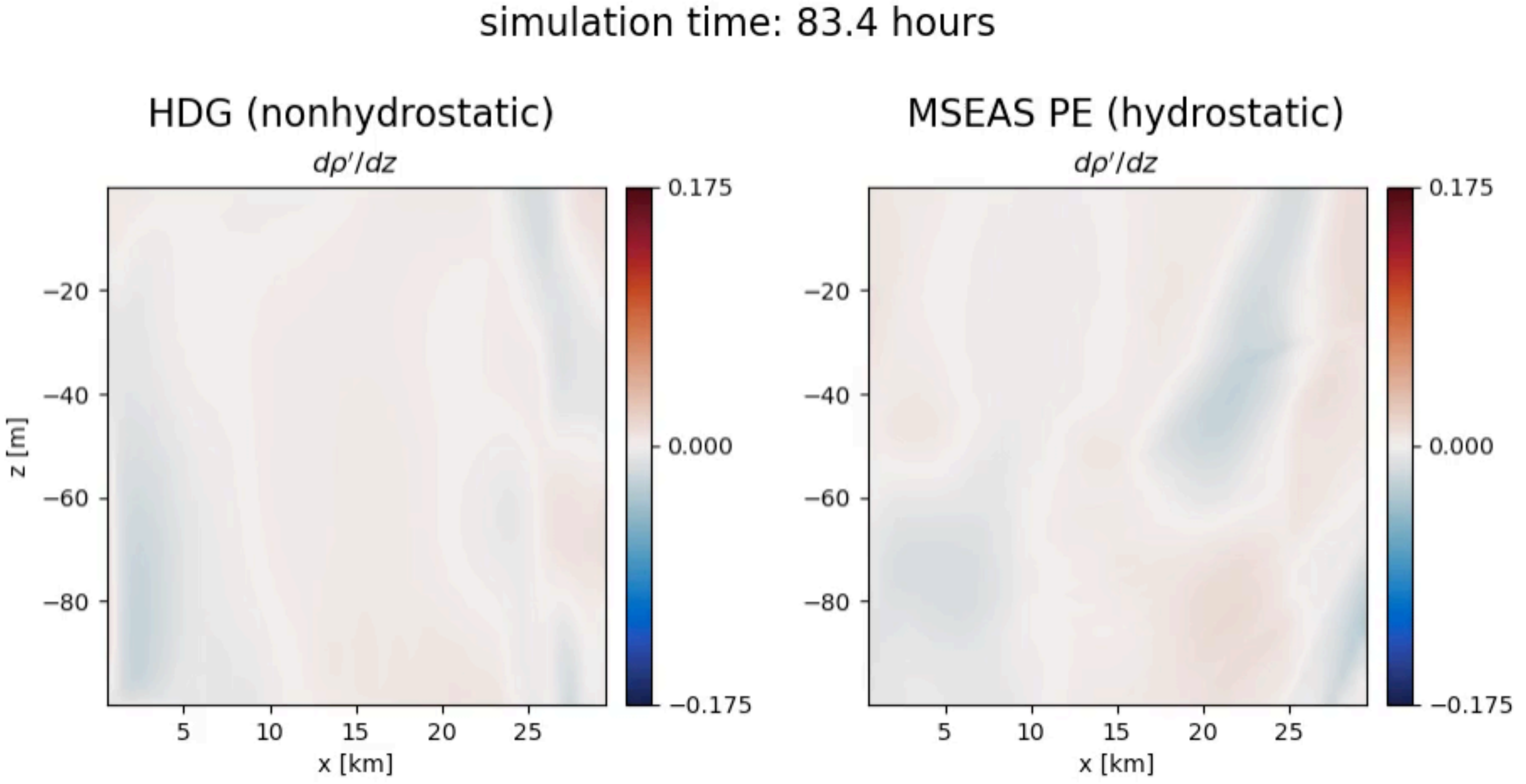
2D model nesting in the Alboran sea



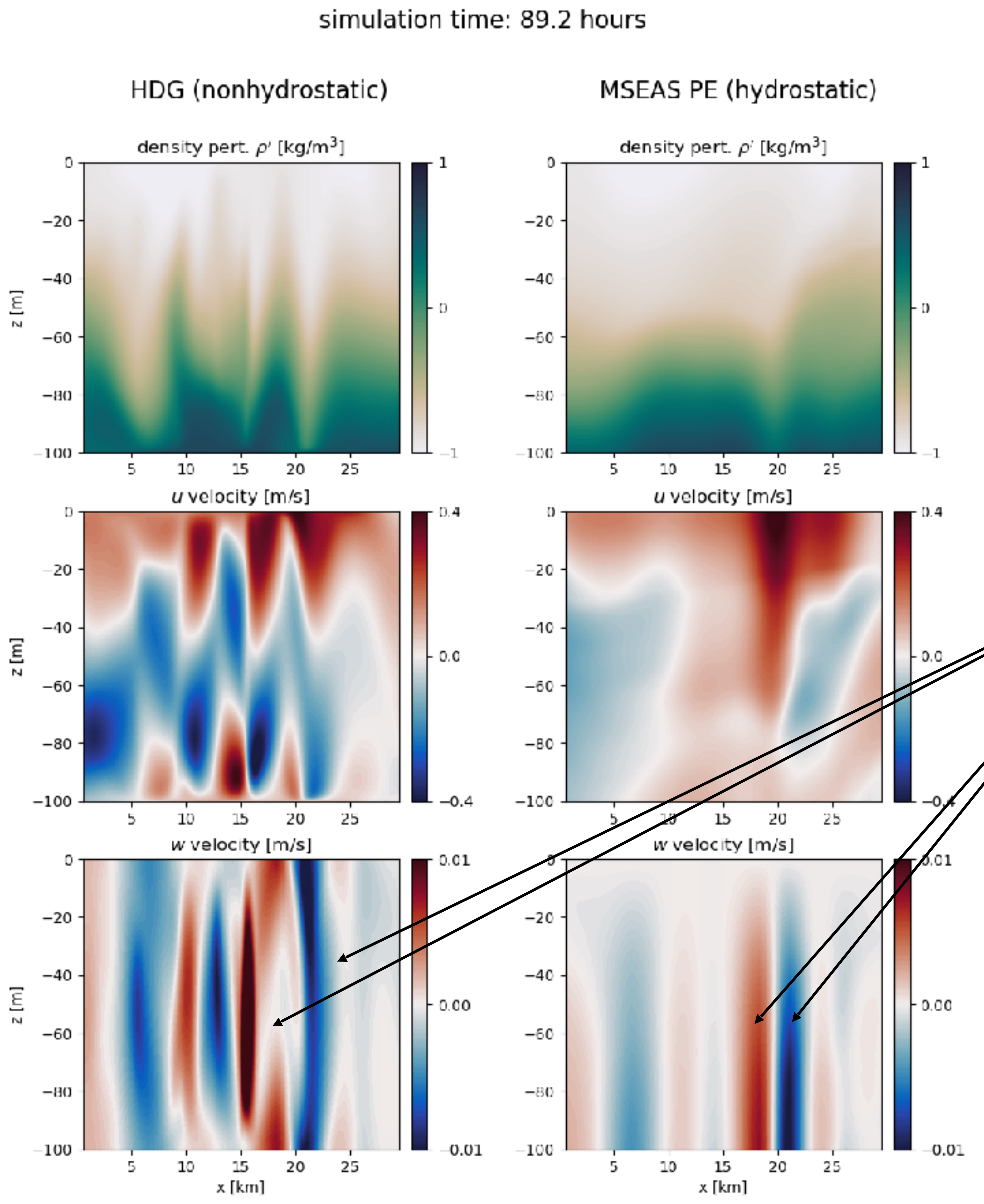


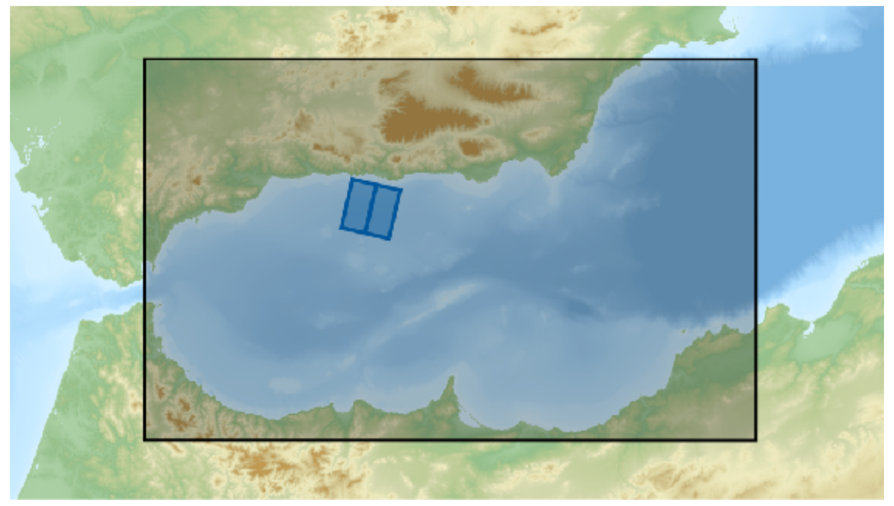
$$\frac{\partial \rho'}{\partial z}$$ **- measure of instability**

red - unstable, blue - stable



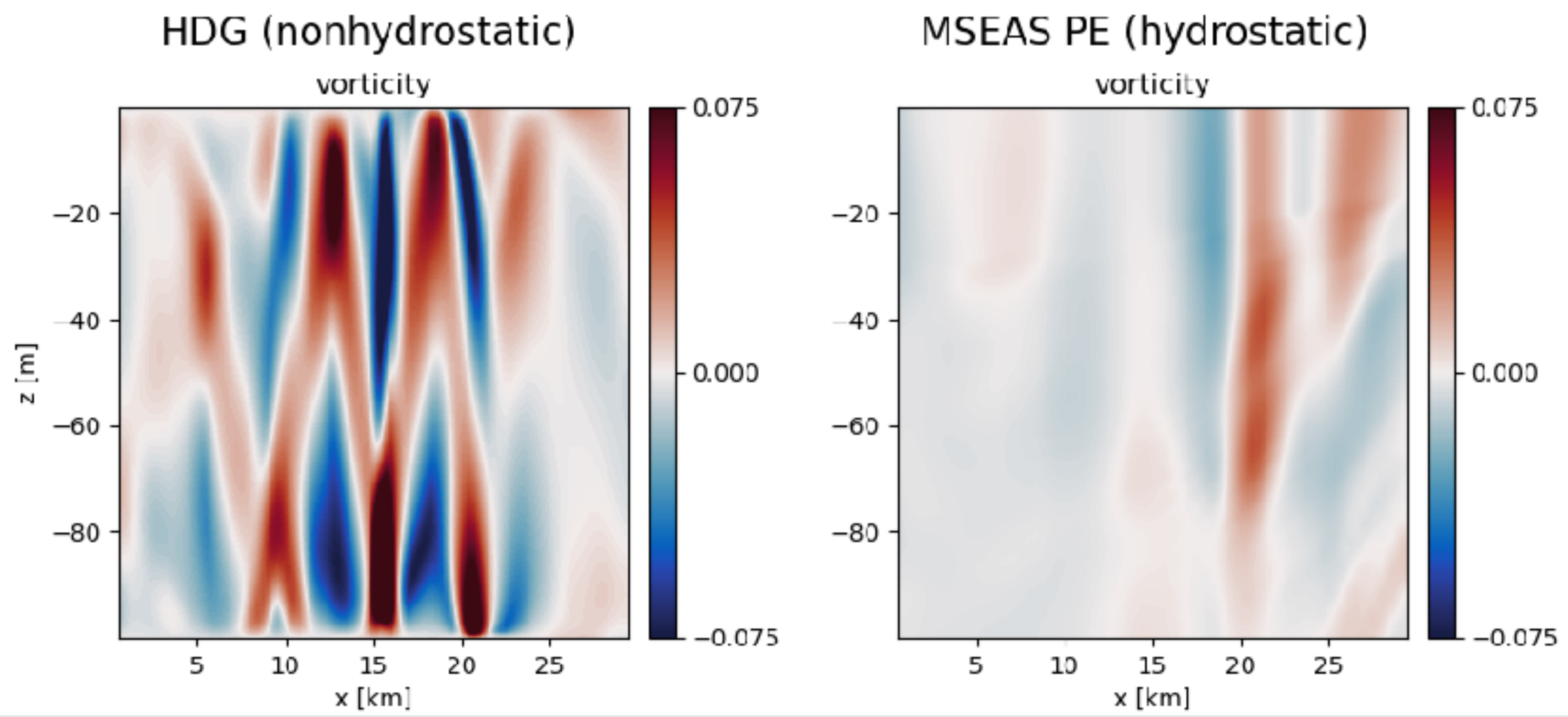
Initialization from Alboran sea data (MSEAS-PE)





During instability:

- about 90 hours into the simulation
- overturning present in the NHS model
- another strong dynamic difference between the two models
- remarkably, the length/time scales of the vertical bands in the velocity are similar for both models
- less "columnar" for the NHS model



[1] **Foucart** et al., Ocean Modelling. in prep. 2023a.

3D Model Nesting in the Alboran Sea

Region of investigation:

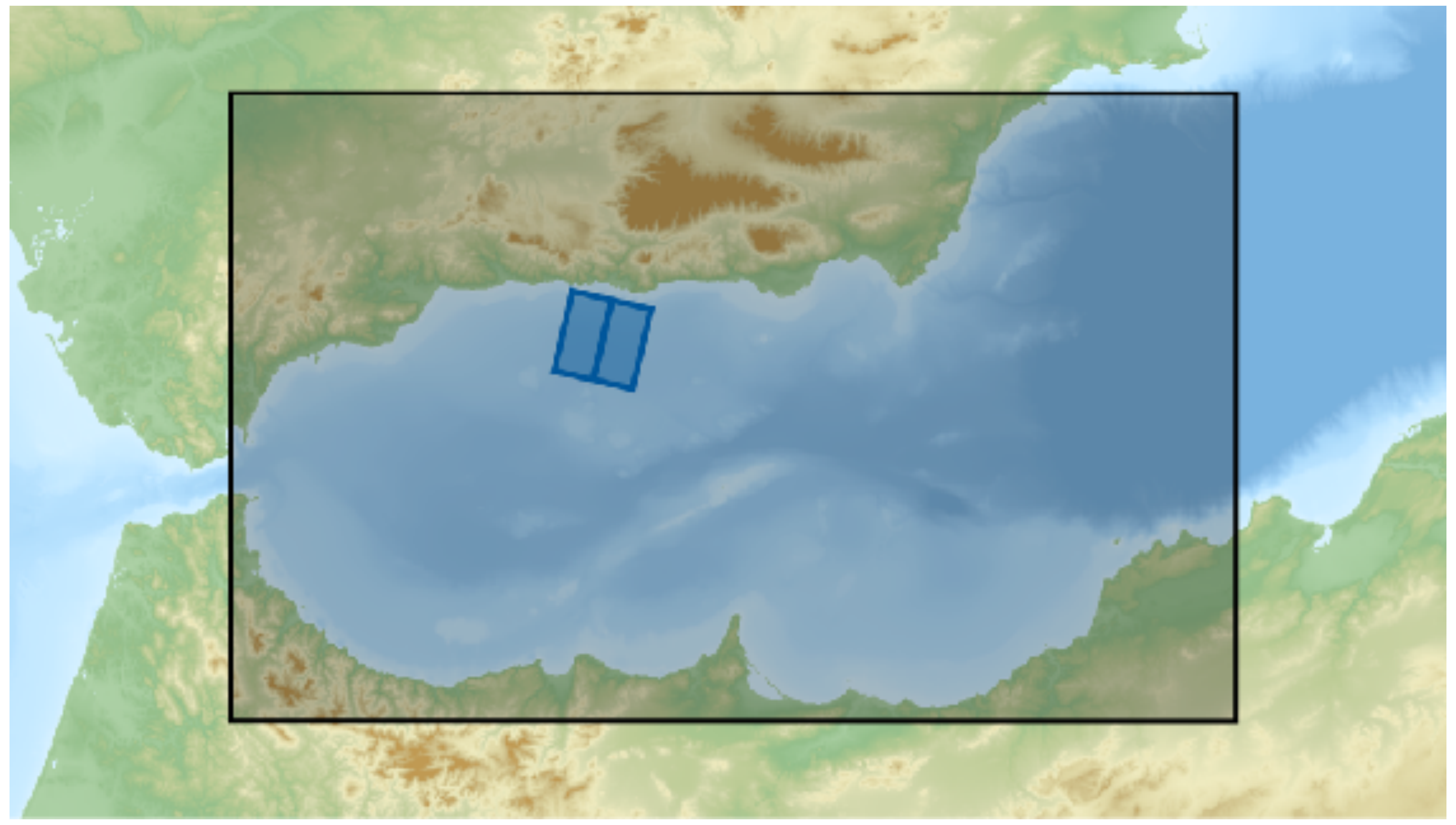
- **Modeling domain:** denotes a section in the Alboran Sea along which **wind-driven instabilities were observed** in the hydrostatic MSEAS PE simulations

- The section starts in the west along the northern edge of the West Alboran Gyre and extends to the east-by-northeast out of the gyre towards the Spanish coast.

- Implemented 3D HDG NHS model nesting and initialization from real data

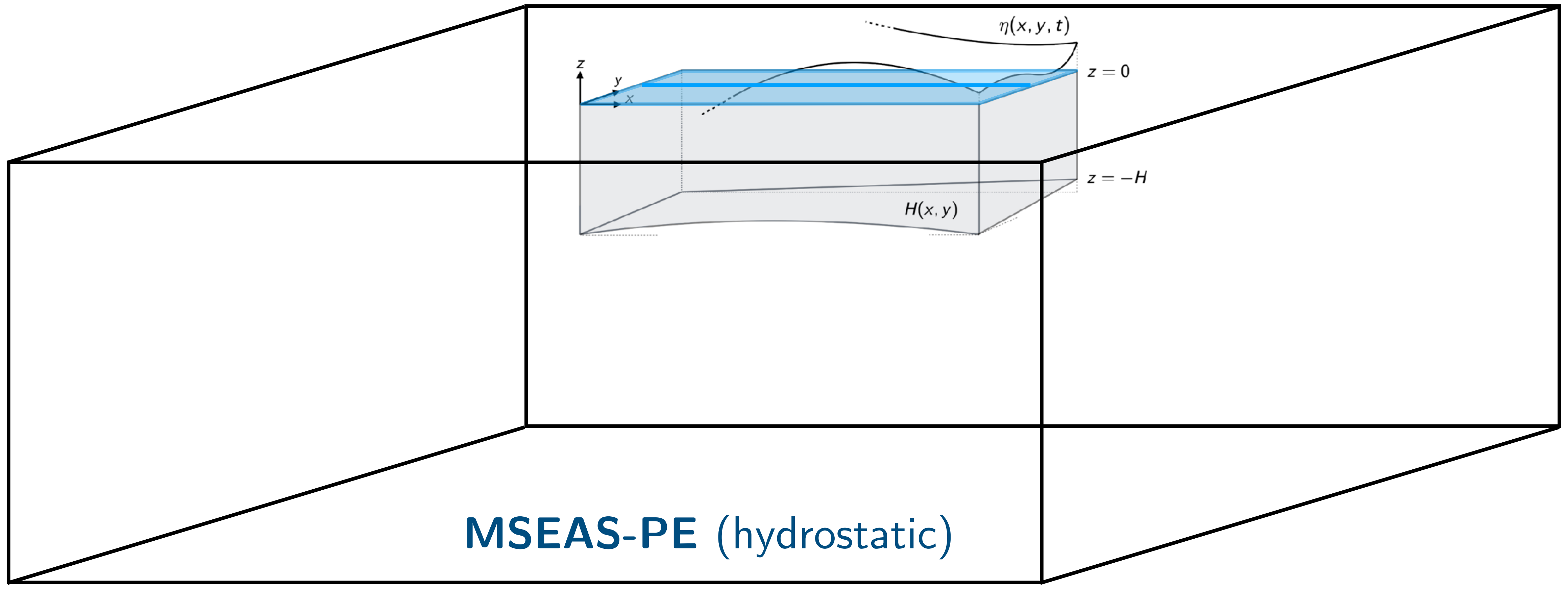
- Goal: model instabilities in the mixed layer and compare to HS simulation output as well

modeling domains



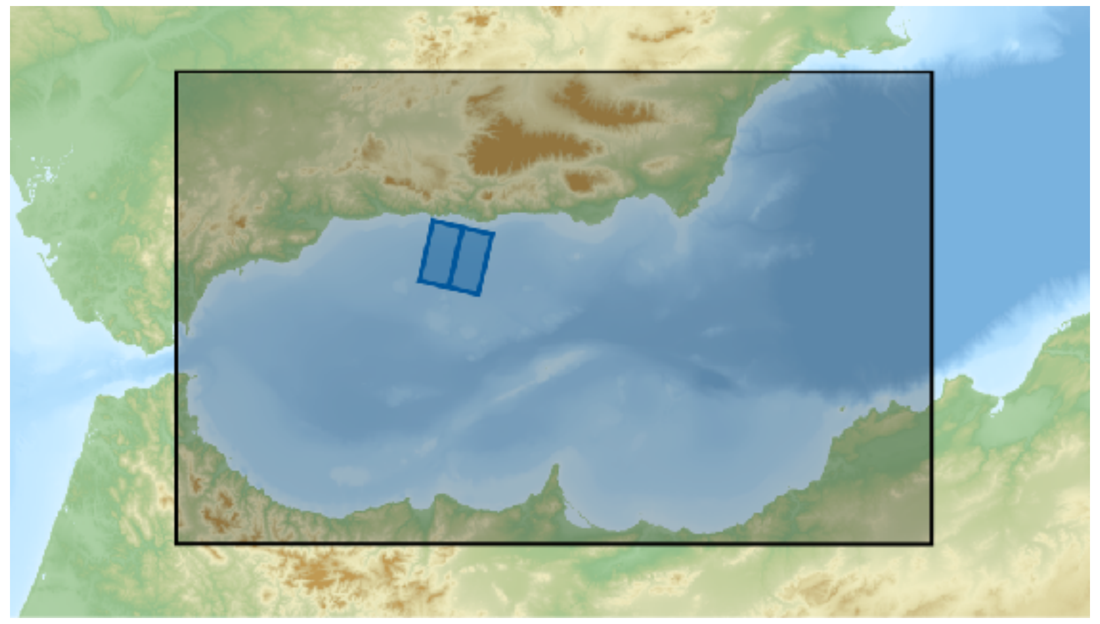
Modeling domain for model nested runs in the Alboran Sea showing the MSEAS-PE domain (shaded, black), and the HDG NHS nested modeling domain (blue) in 2D (center slice) and 3D (box).

(not to scale)

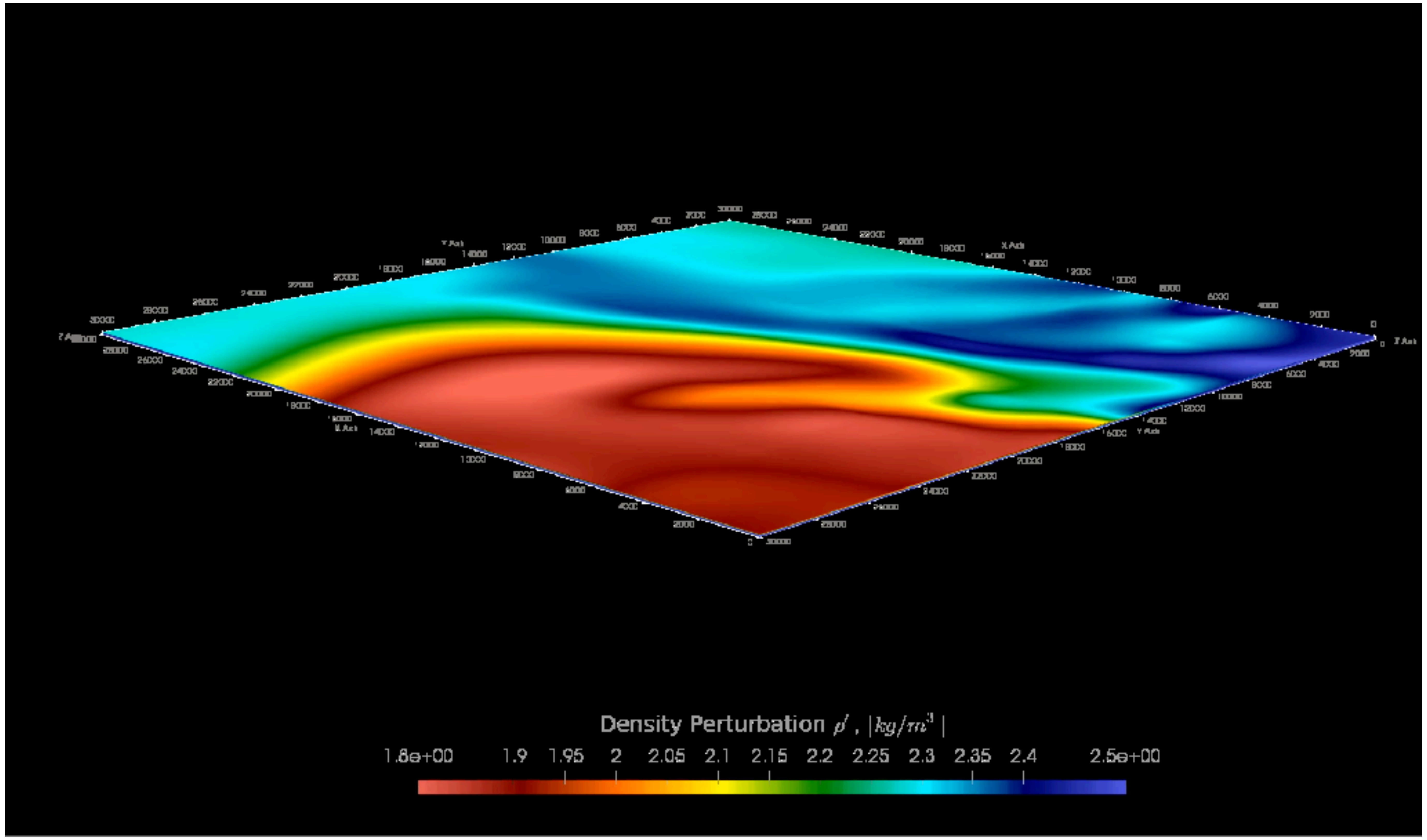


[1] **Foucart** et al., Ocean Modelling. in prep. 2023a.

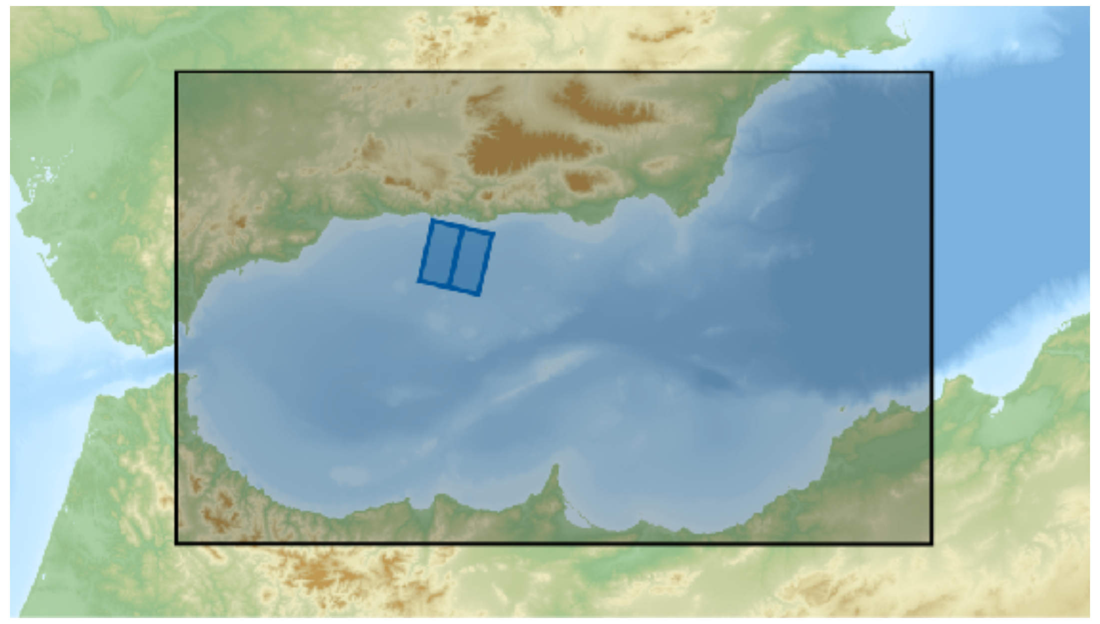
3D model nesting in the Alboran sea data (MSEAS-PE)



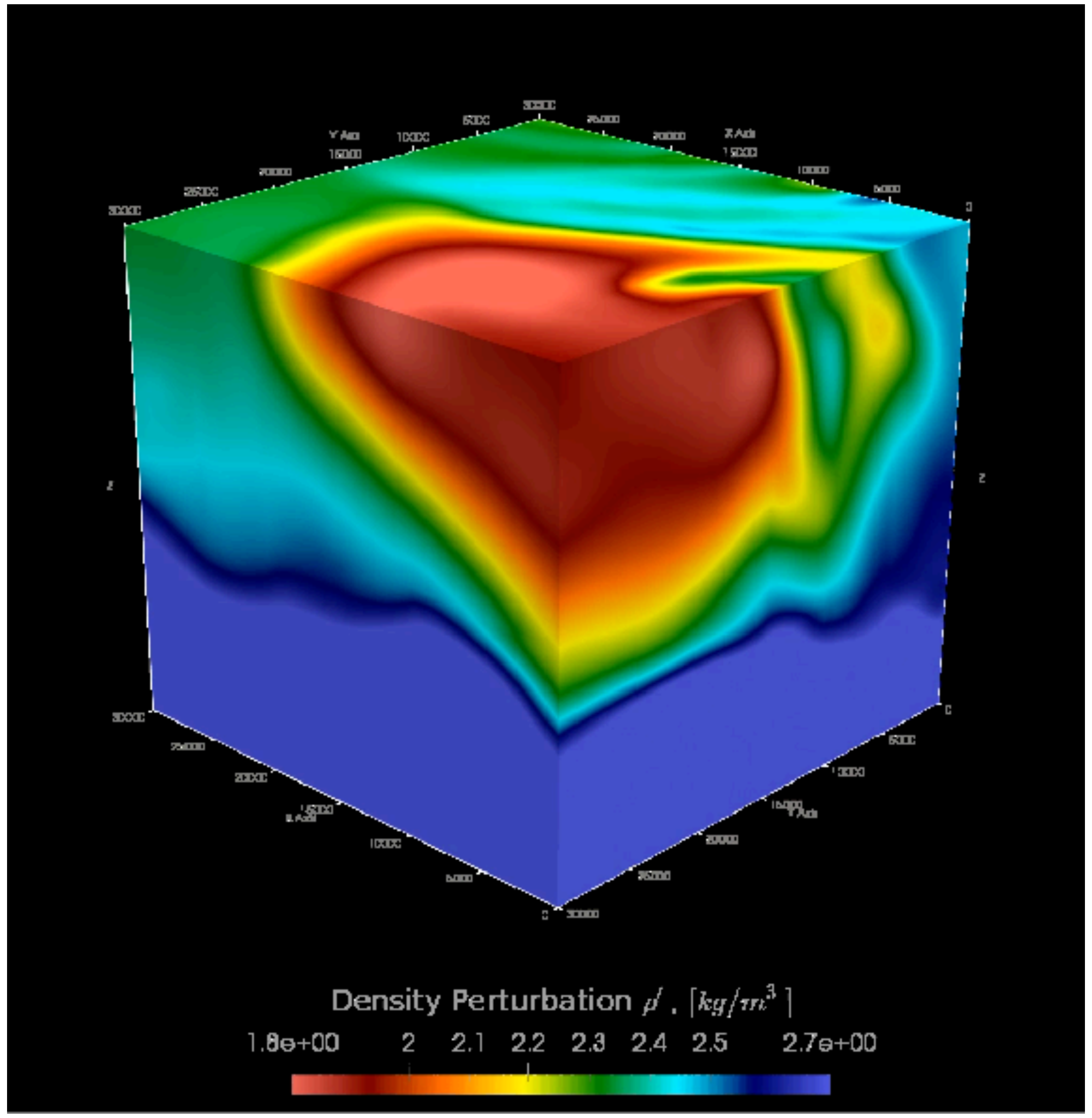
Density perturbation, domain to scale

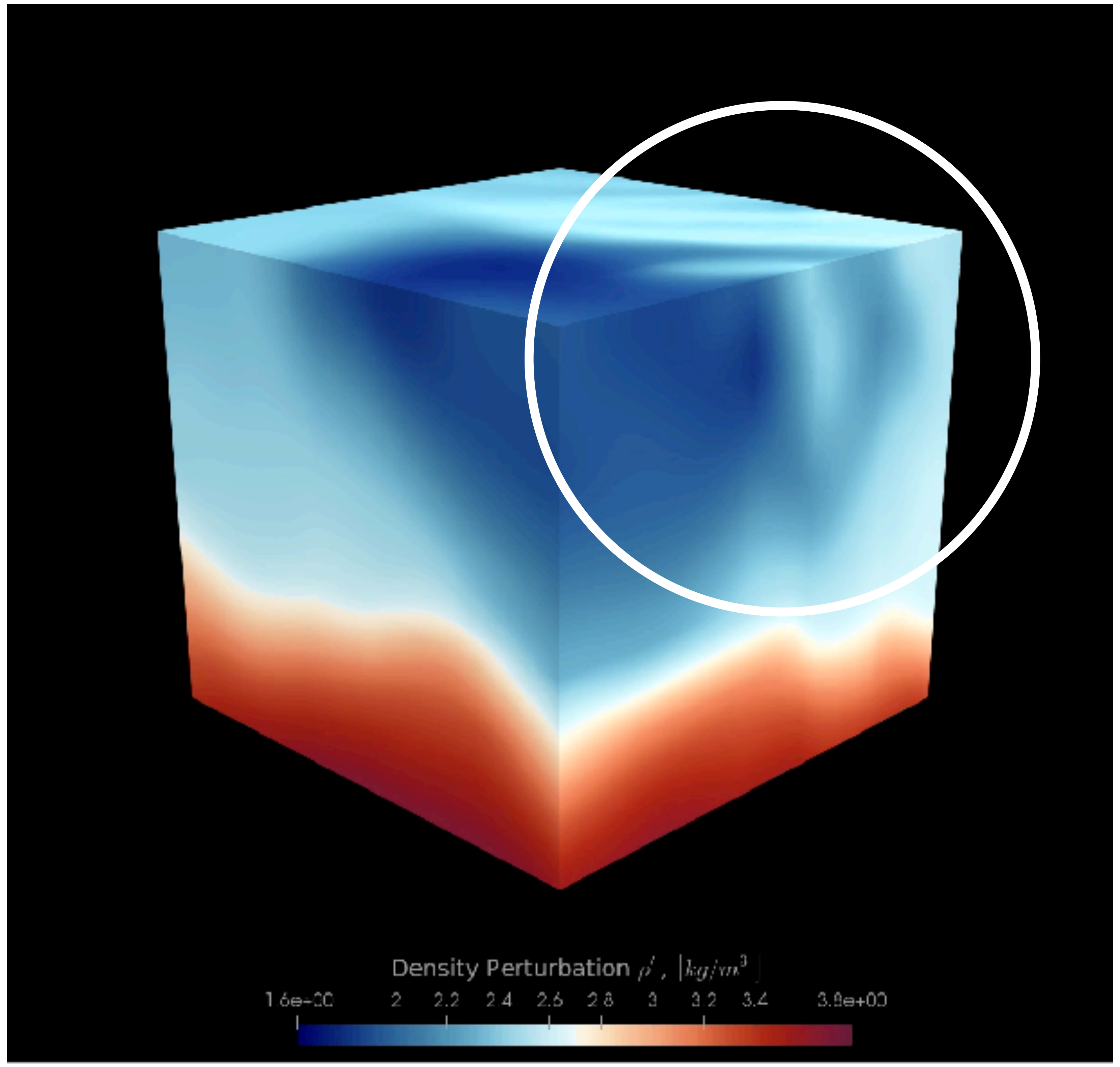


3D model nesting in the Alboran sea data (MSEAS-PE)



Domain scaled by factor of 300 in z

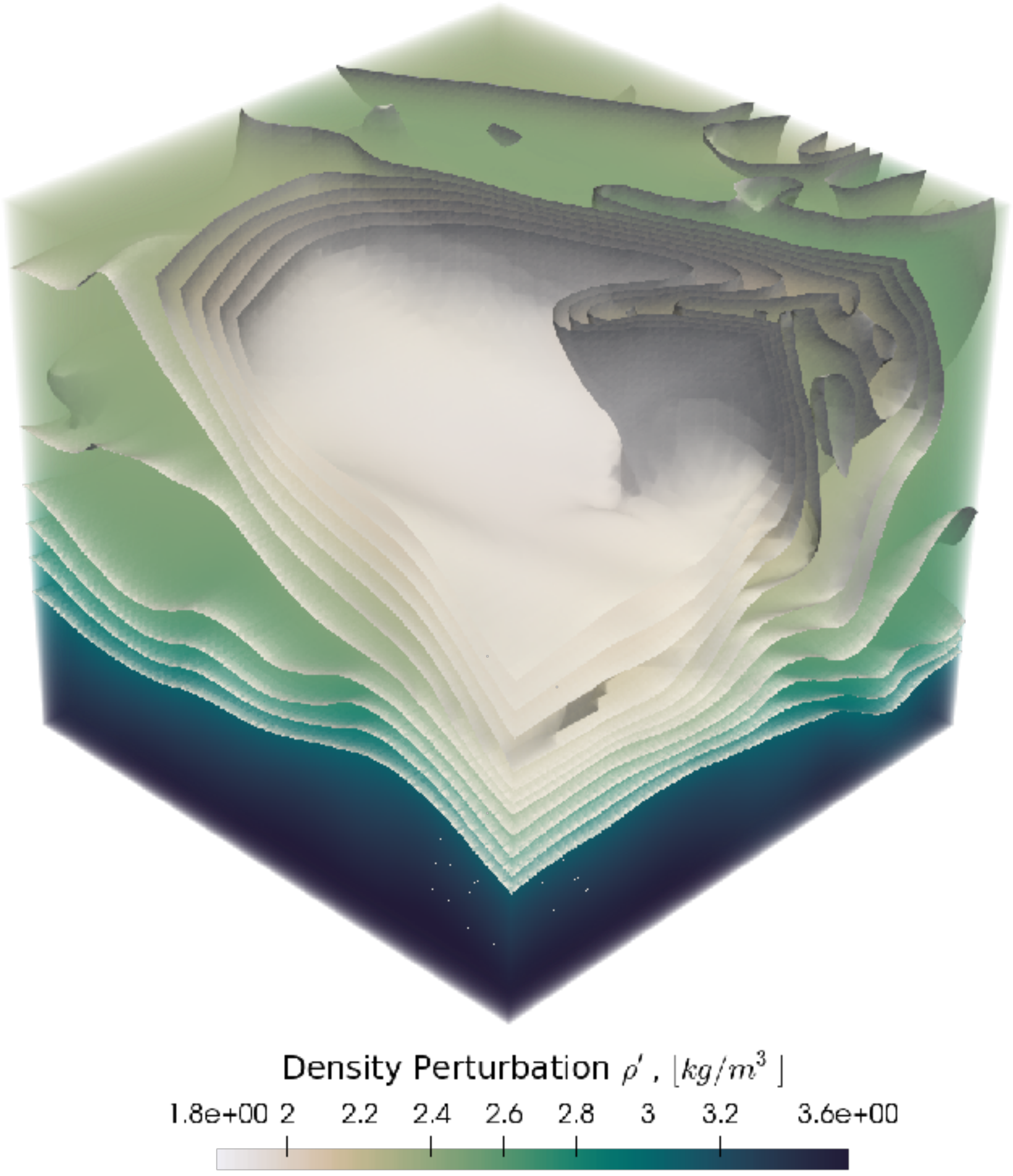


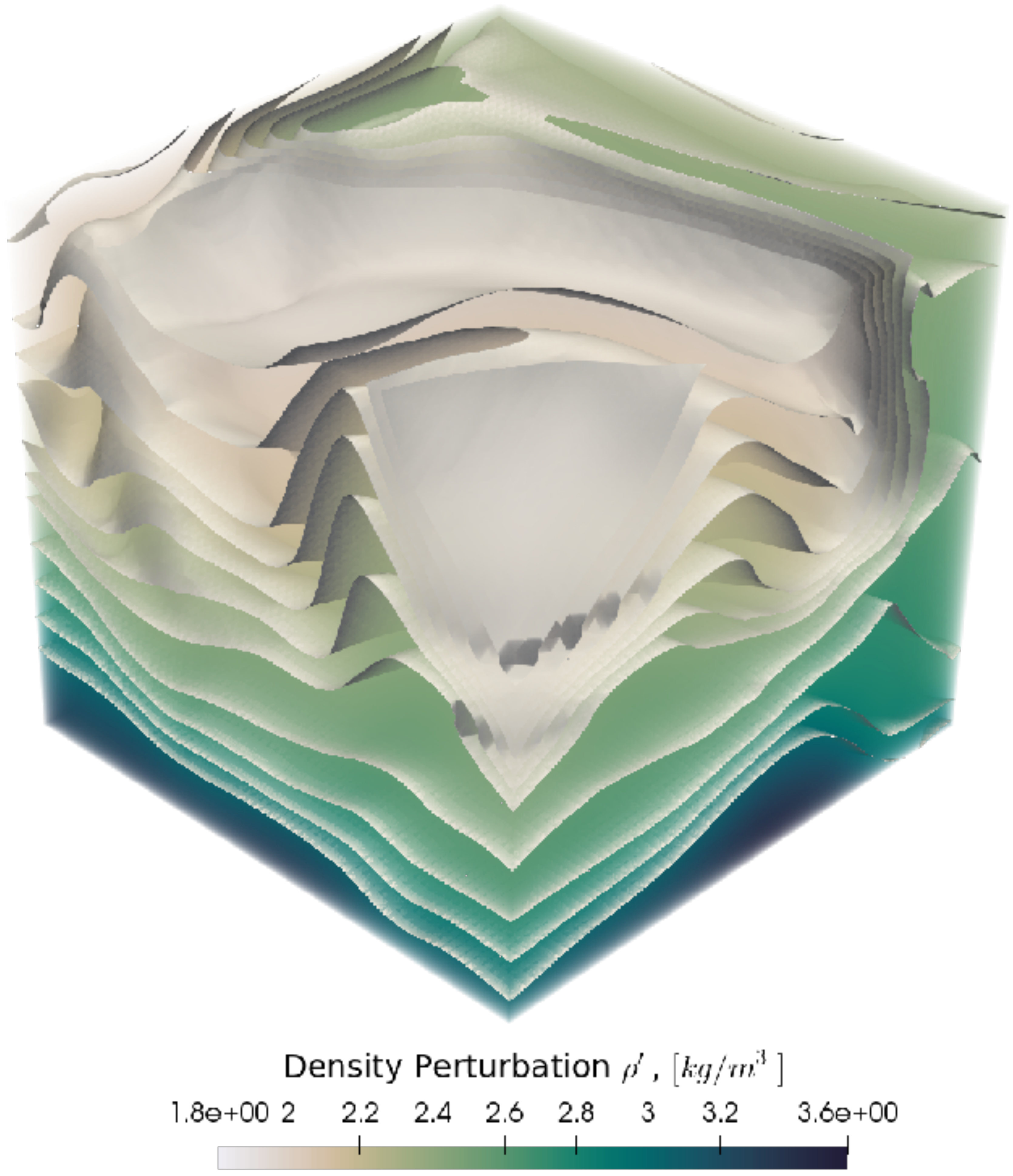


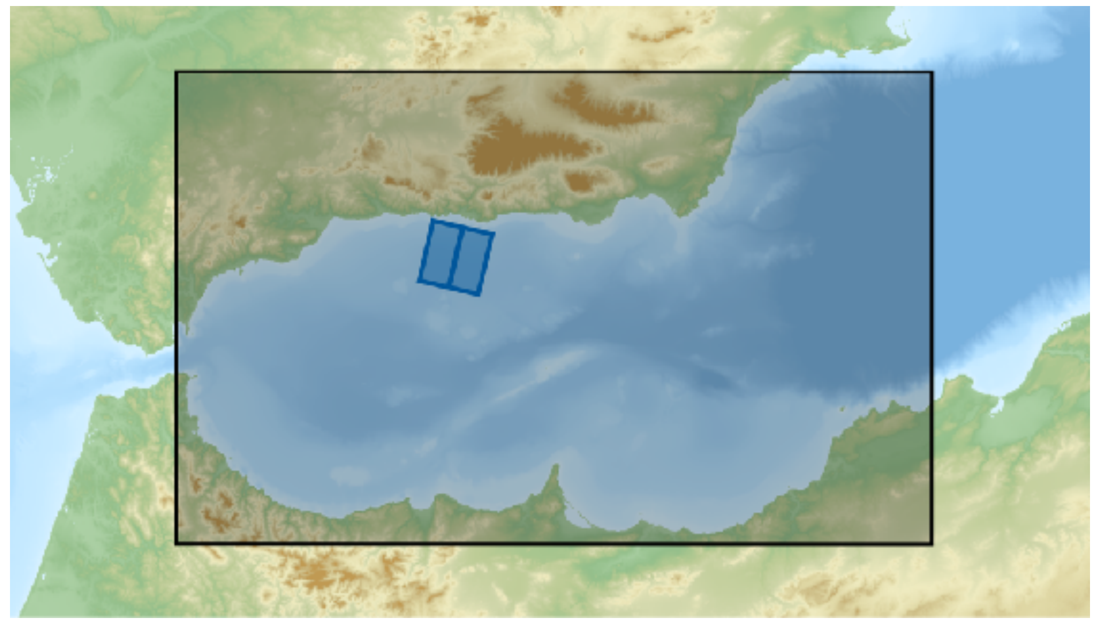
- In 3D, interpretation can be very sensitive to the visualization

[1] **Foucart** et al., Ocean Modelling. in prep. 2023a.

3D model nesting in the Alboran sea data (MSEAS-PE)



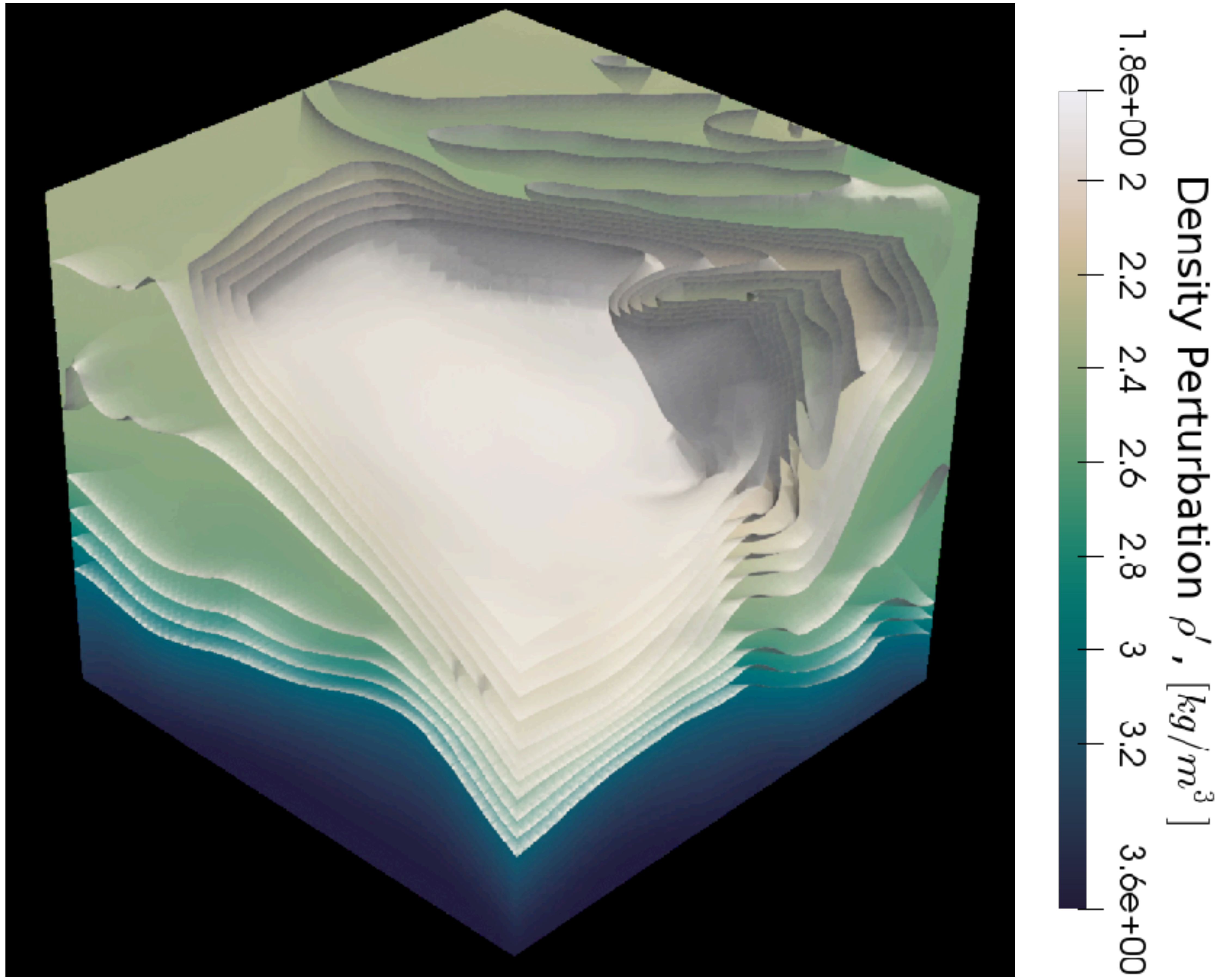


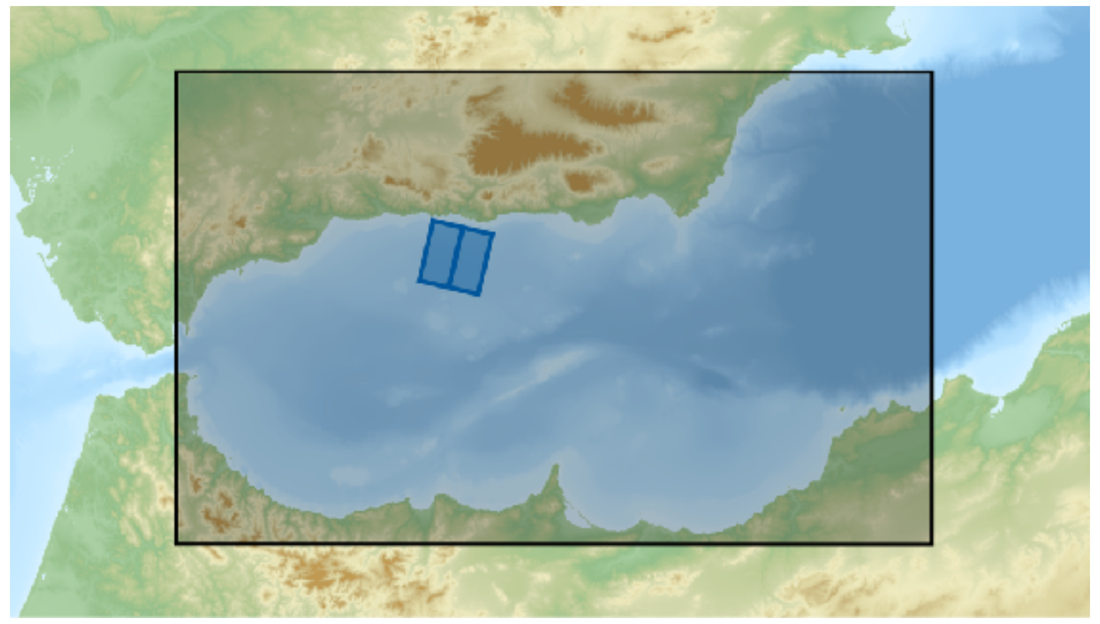


[1] **Foucart** et al., Ocean Modelling. in prep. 2023a.

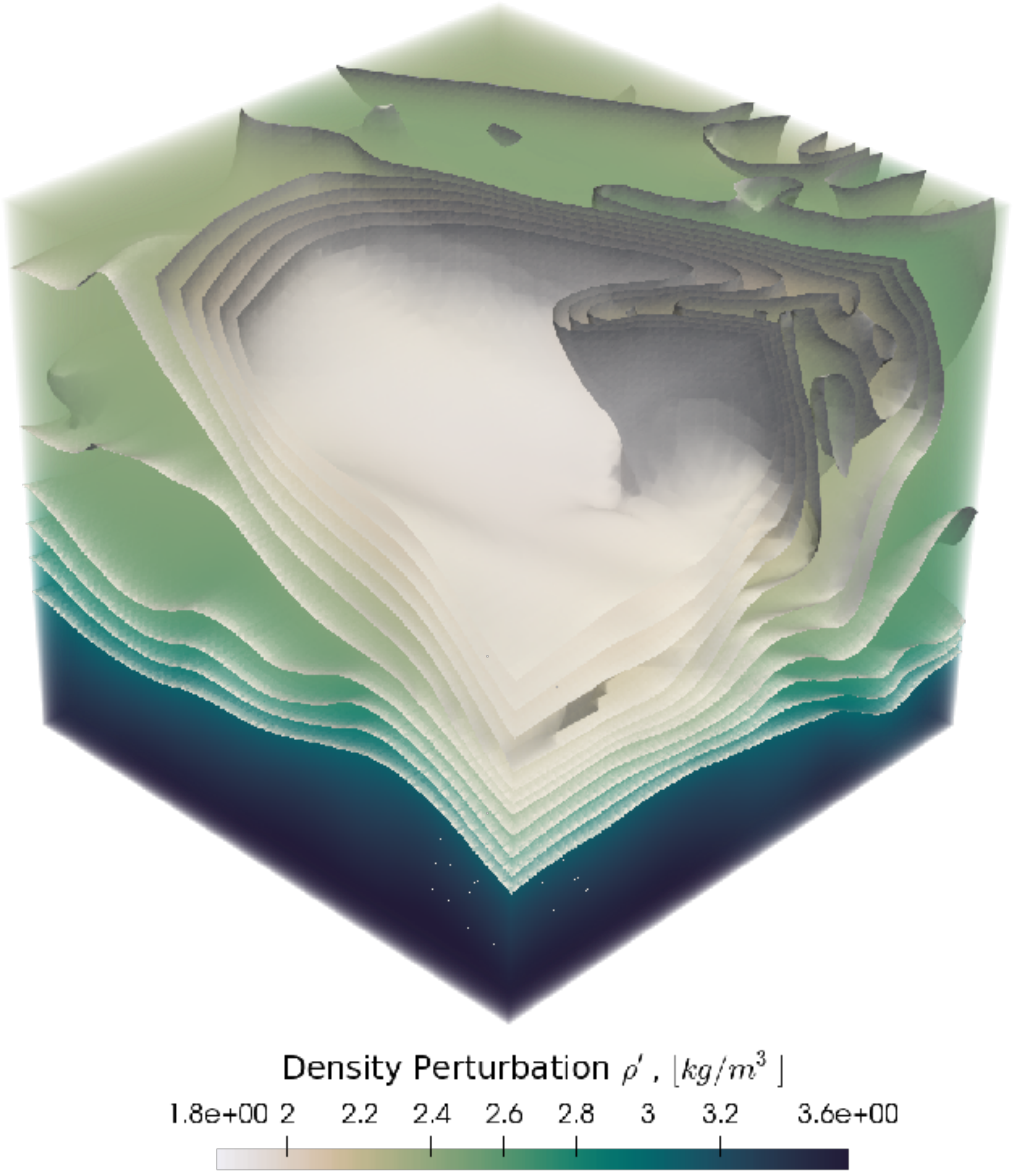
3D model nesting in the Alboran sea data (MSEAS-PE)

Density perturbation contours, volumetric rendering

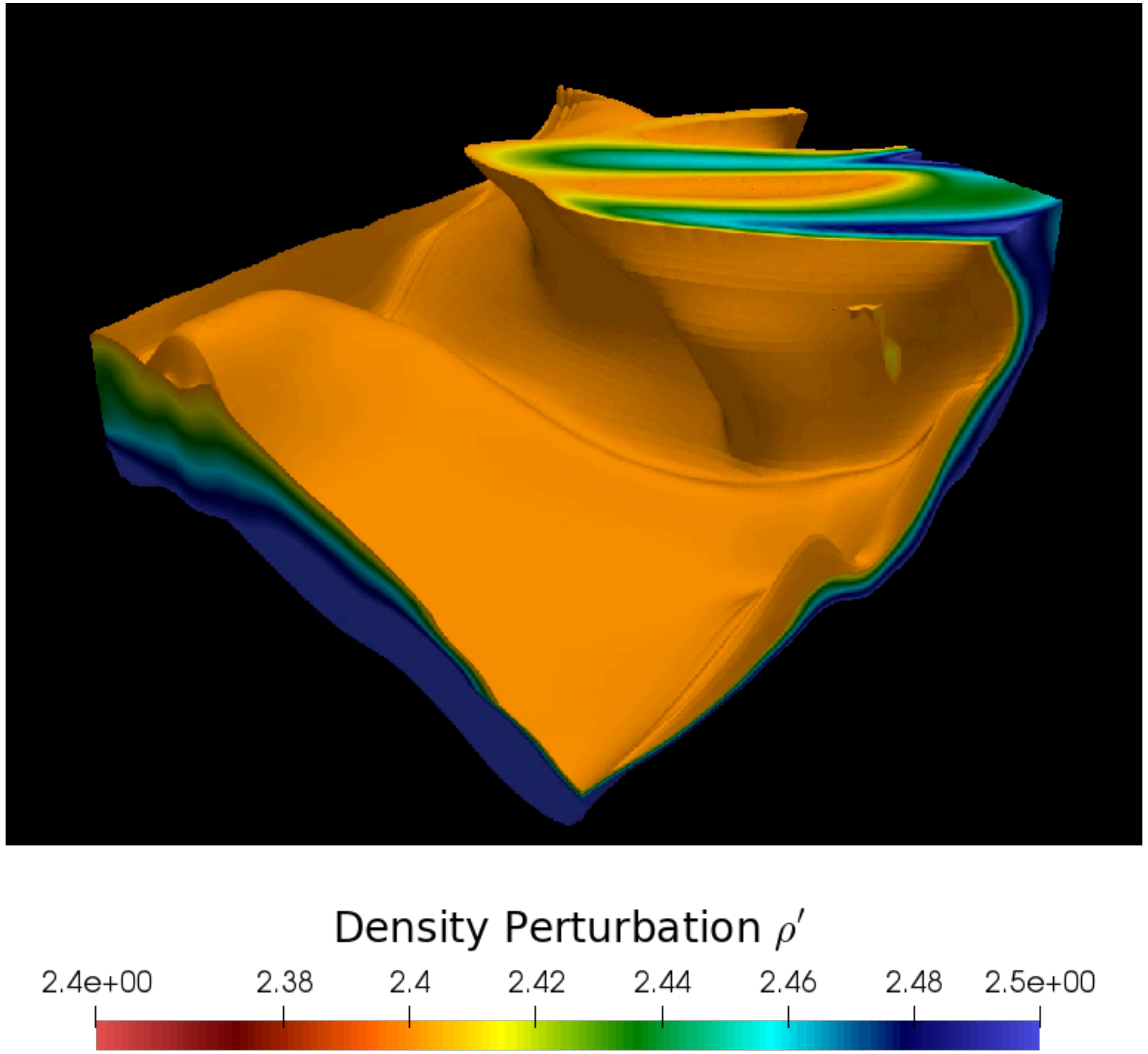




3D model nesting in the Alboran sea data (MSEAS-PE)

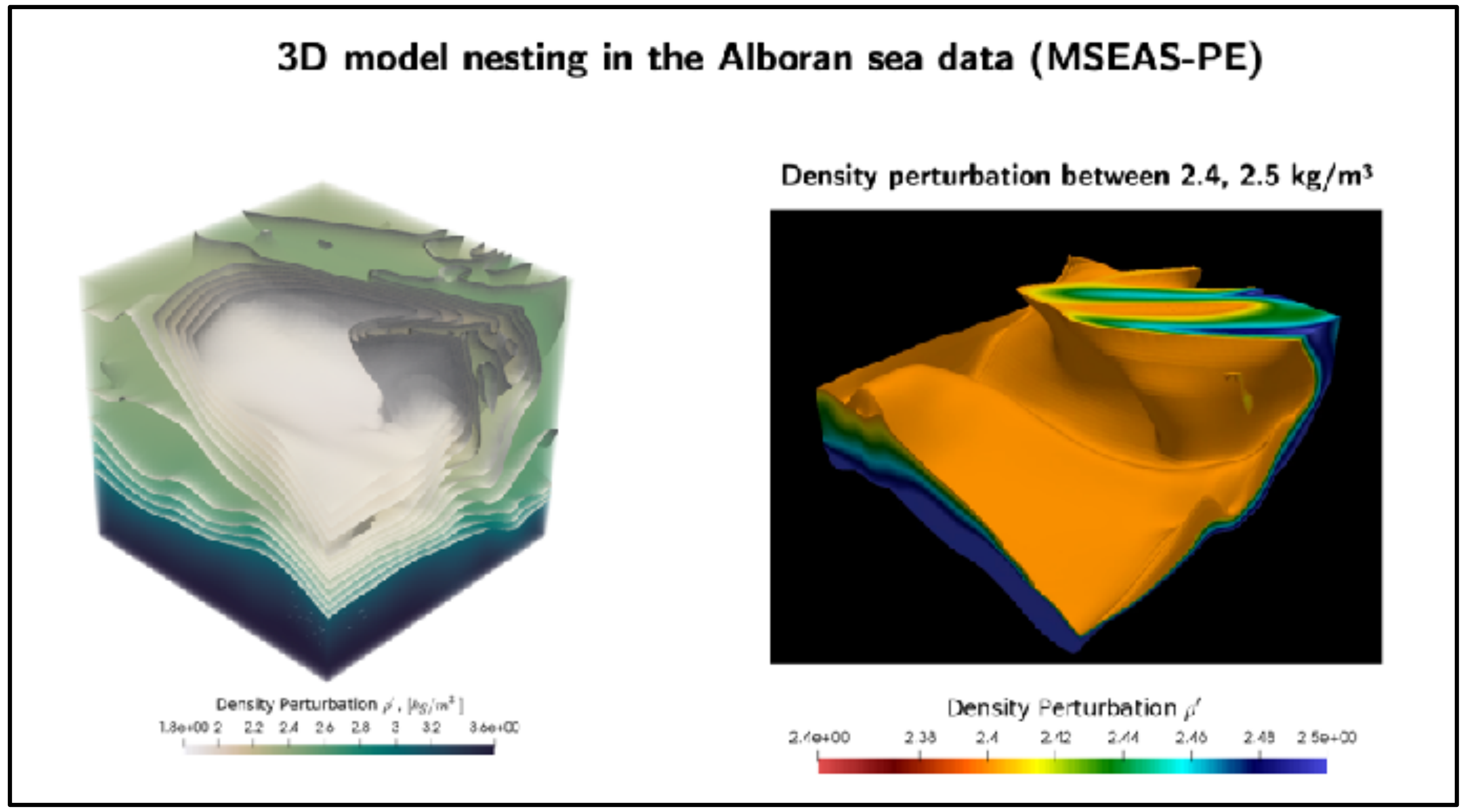


Density perturbation between 2.4, 2.5 kg/m³

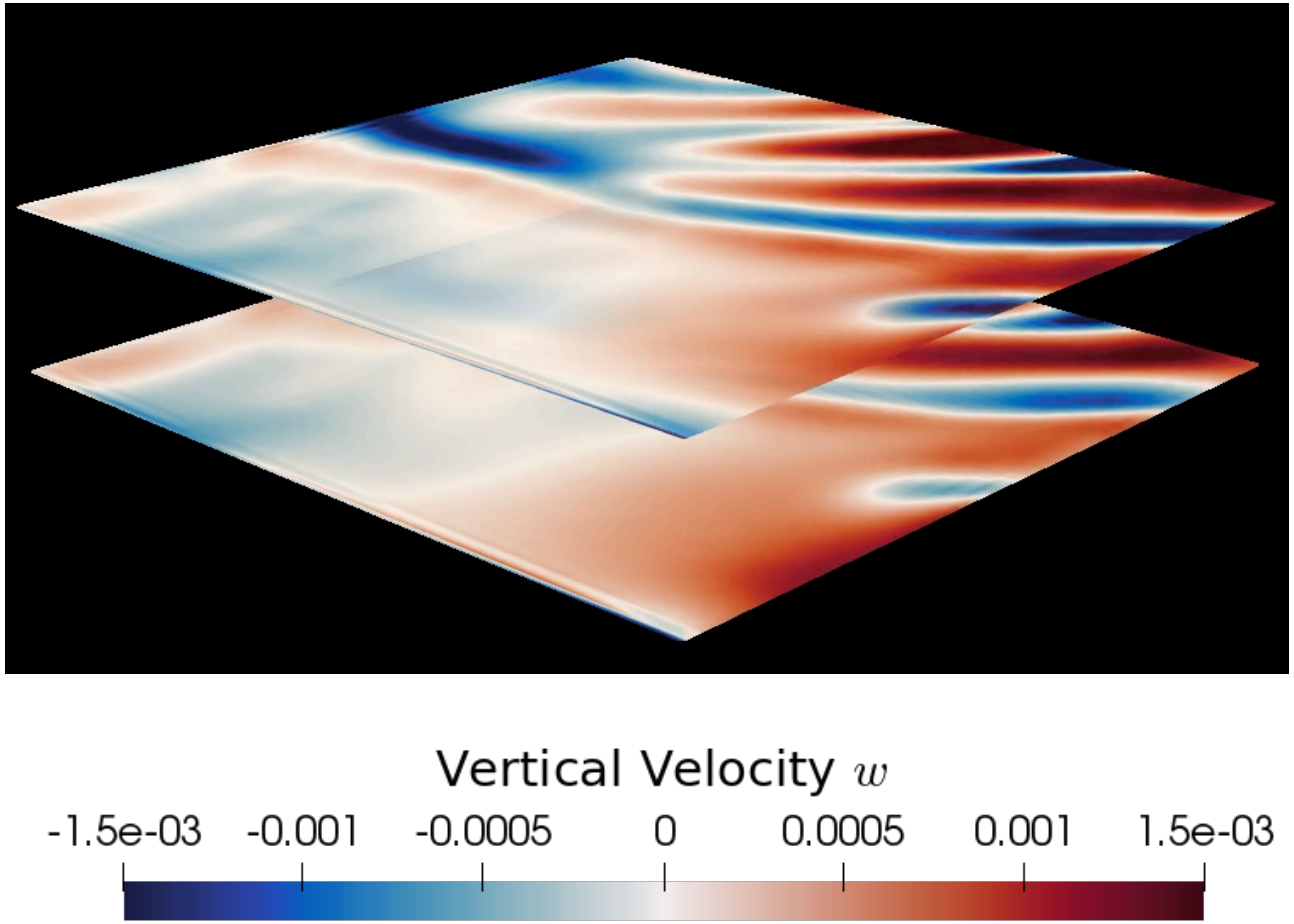


3D model nesting in the Alboran sea data (MSEAS-PE)

Overhang instability

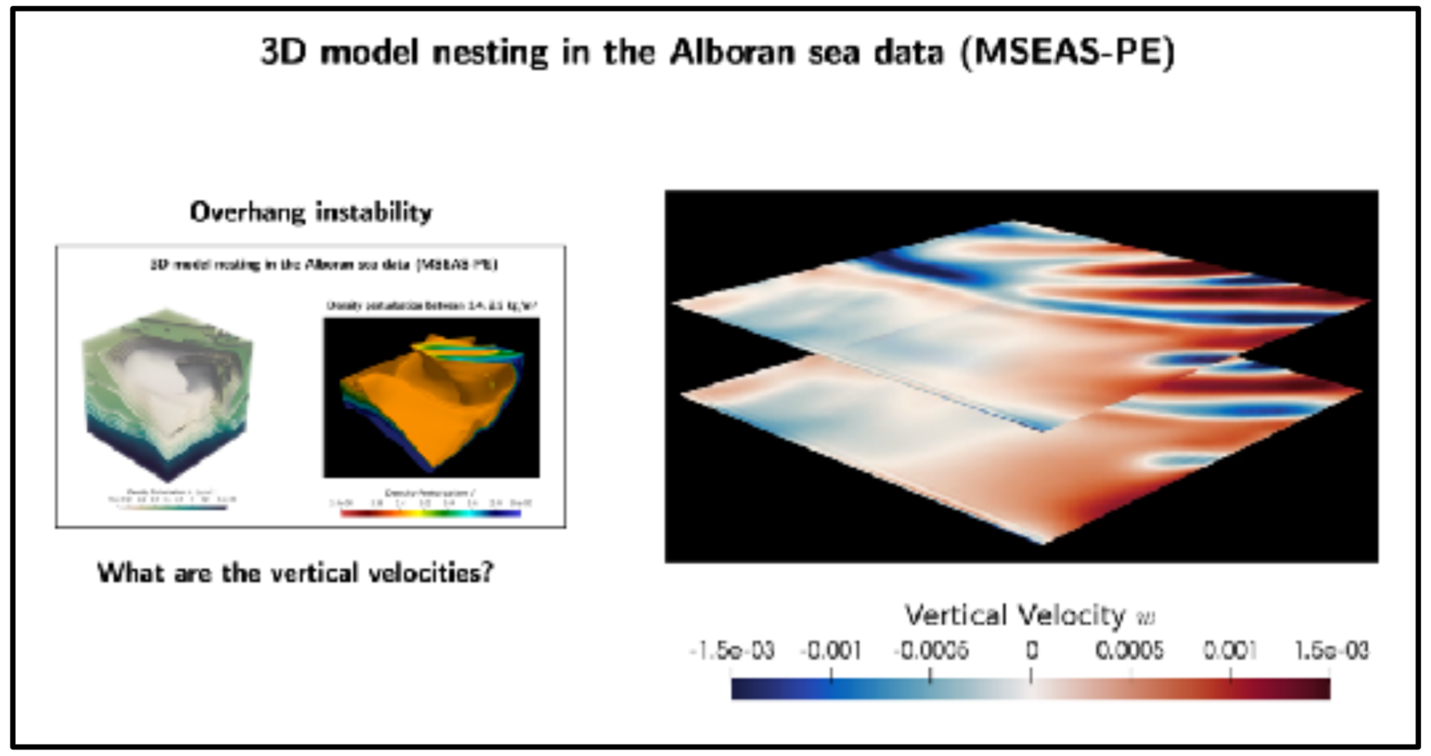


What are the vertical velocities?

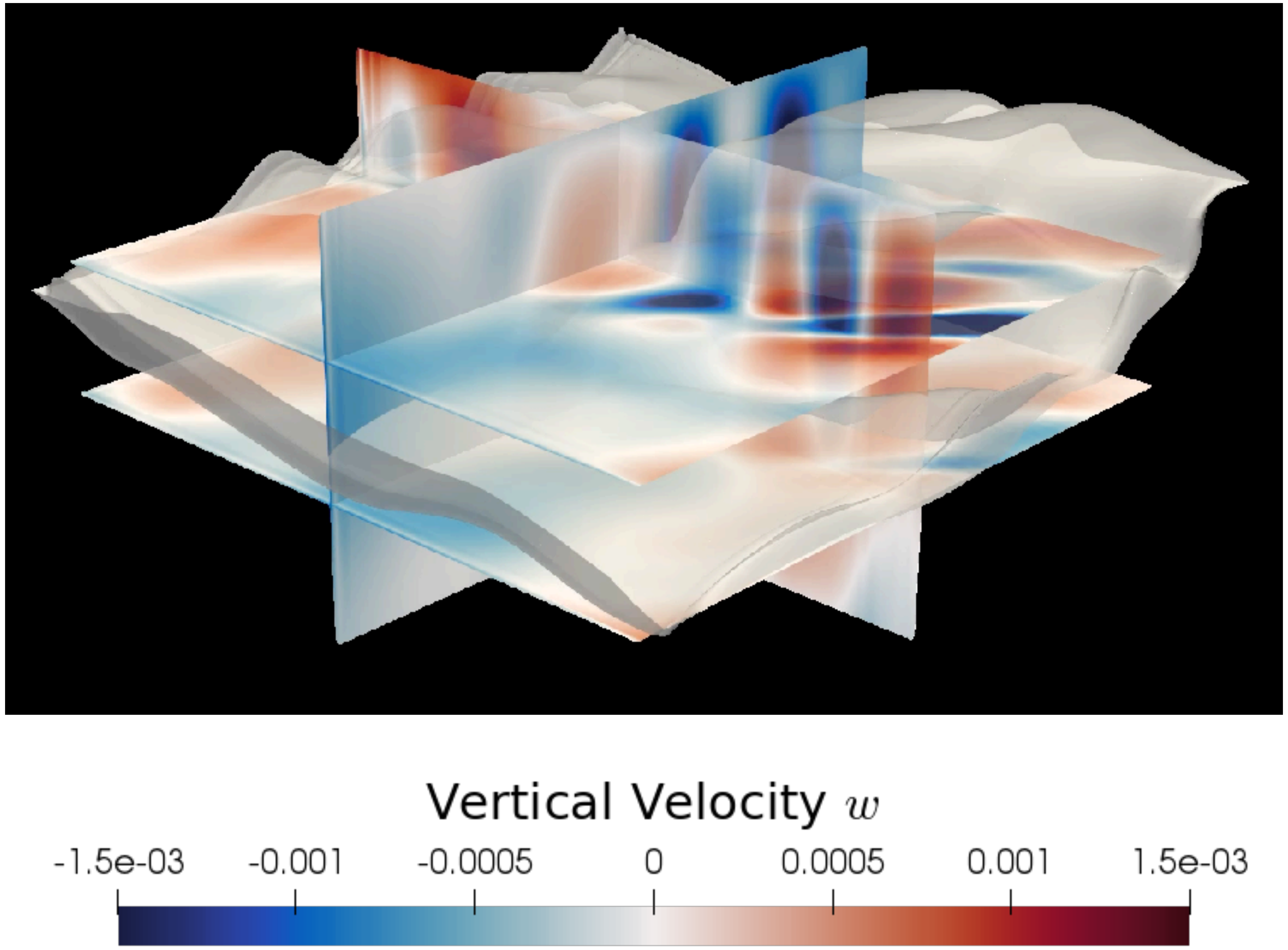


[1] **Foucart** et al., Ocean Modelling. in prep. 2023a.

3D model nesting in the Alboran sea data (MSEAS-PE)

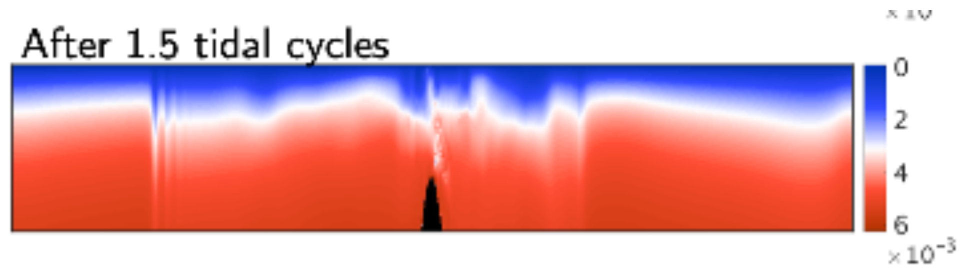


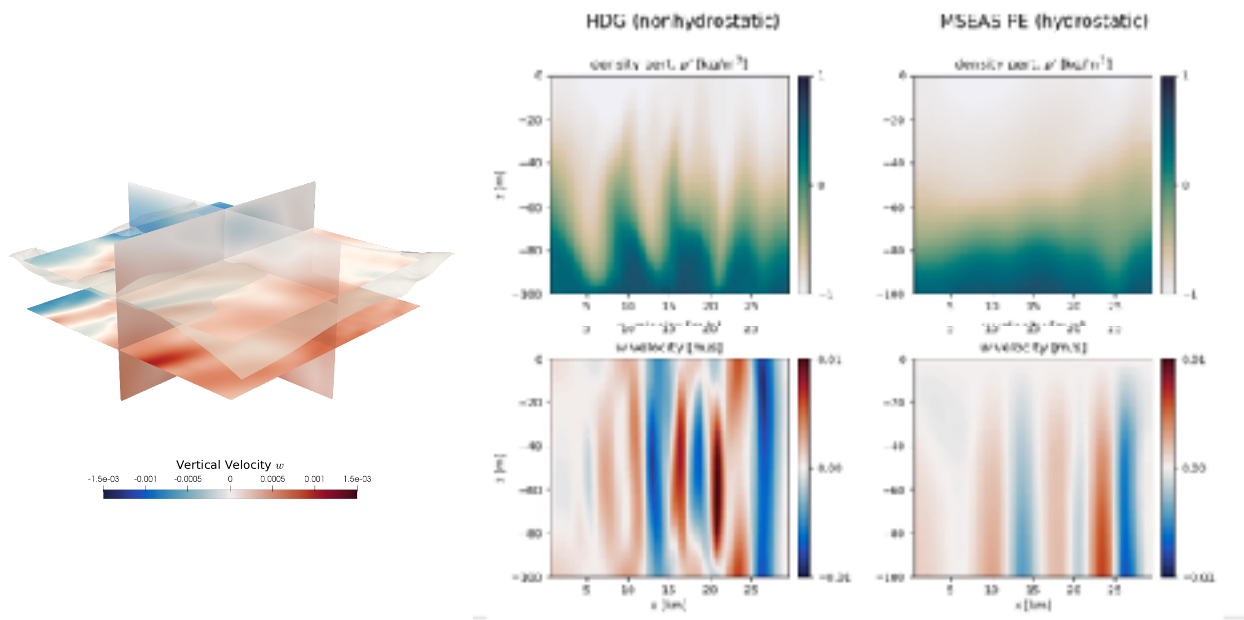
Ideally, we would like to visualize both together

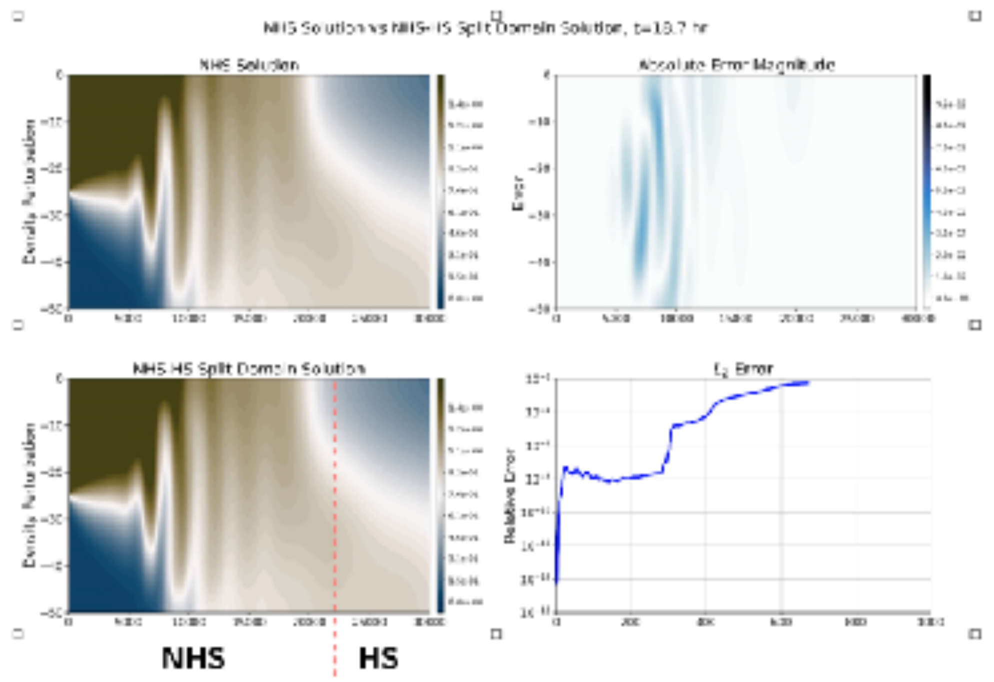


What does this model let us do?

- Perform zonal investigations of nonhydrostatic behavior
  - Areas with steep bathymetry [1], [5], [6]
  - Regions experiencing wind stress [1], [3], [4]
- Validate hydrostatic models
  - Determine when hydrostatic models are capable of resolving the correct length/time scales of ocean processes, and when they aren't [1], [5], [7-9]
- Discover parametrization for incorporation into large legacy hydrostatic models / climate models
  - Statistical descriptions of the same (Oceananigans.jl) [7]
- Hydrostatic/nonhydrostatic domain splitting
  - Work already started in our group [9]
  - DG/HDG boundary conditions advantageous for such approaches







[1] **Foucart** et al., Ocean Modelling. in prep. 2023a. [2] Freilich & Mahadevan, JGR, 2021 [3] Calil, OD, 2017 [4] Kaempf, OD, 2019 [5] Vitousek and Fringer 2014 [6] Buijsman, Kanarska, and McWilliams 2010 [7] Ramadhan, et. al (2022) [8] Foucart et al., 2018 [9] A. Karthik, MSEAS, SM Thesis

Outline

Deep reinforcement learning for adaptive mesh refinement

Motivation:

- machine learning excels at learning latent patterns from large datasets in the absence of a model [1]
- a PDE is often an excellent model, and numerical methods provide stability, consistency, and convergence guarantees
- using machine learning to directly learn solutions to PDEs is subject to demonstrable failure modes, and often generalizes poorly [2]

Idea:

- use machine learning to improve heuristic elements of numerical solvers for which we don't have a good model
- adaptive mesh refinement is one such aspect

[1] Bishop, 2006 [2] Krishnapriyan et al., 2021 [3] Foucart et al., JCP. 2022.

Adaptive mesh refinement (AMR)

PDE:

$$\nabla \cdot (\mathbf{c}u) = f$$

$$u = g_D \qquad \text{on } \Gamma_{\text{inflow}}$$

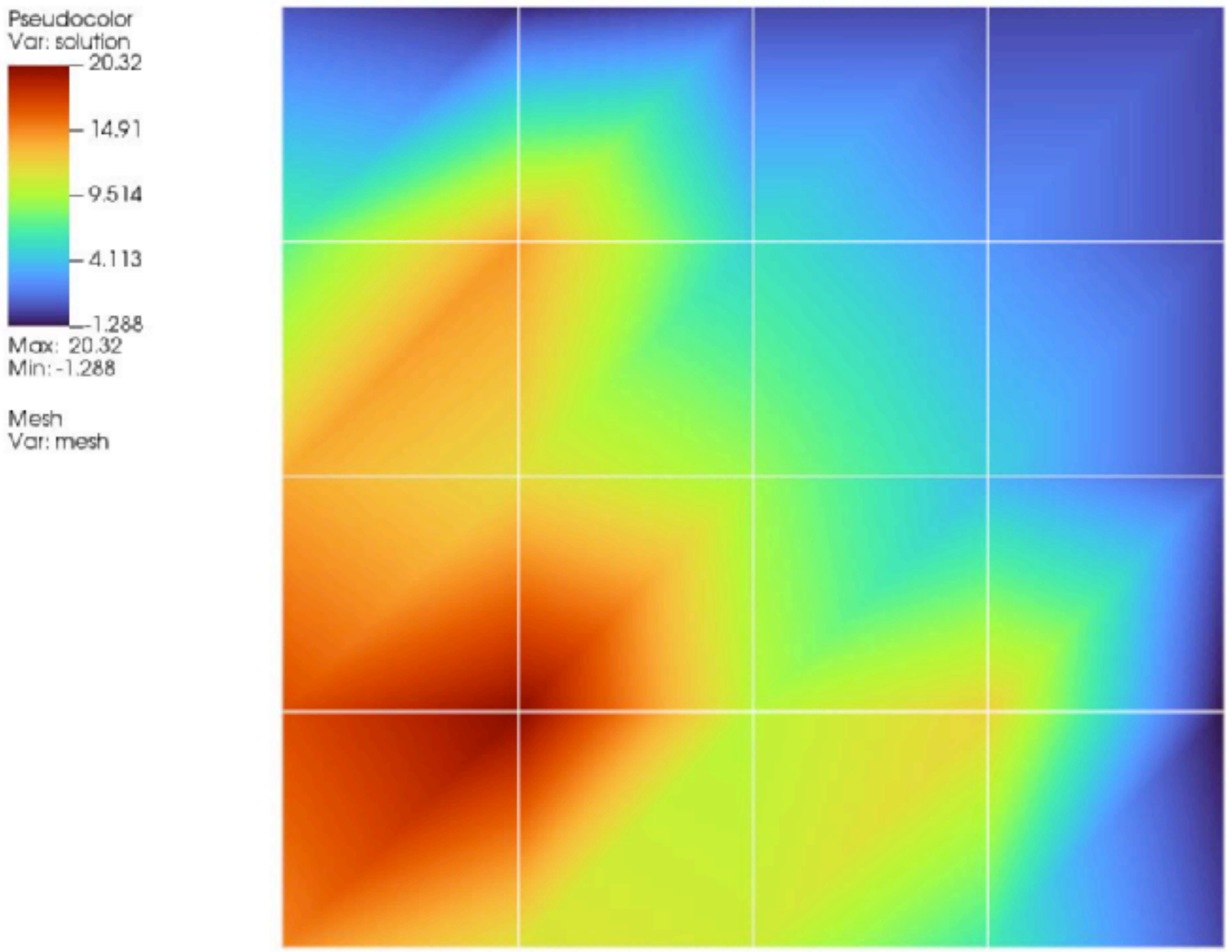
Idea: refine mesh only "where something is happening"

AMR strategies:

SOLVE ESTIMATE MARK REFINE

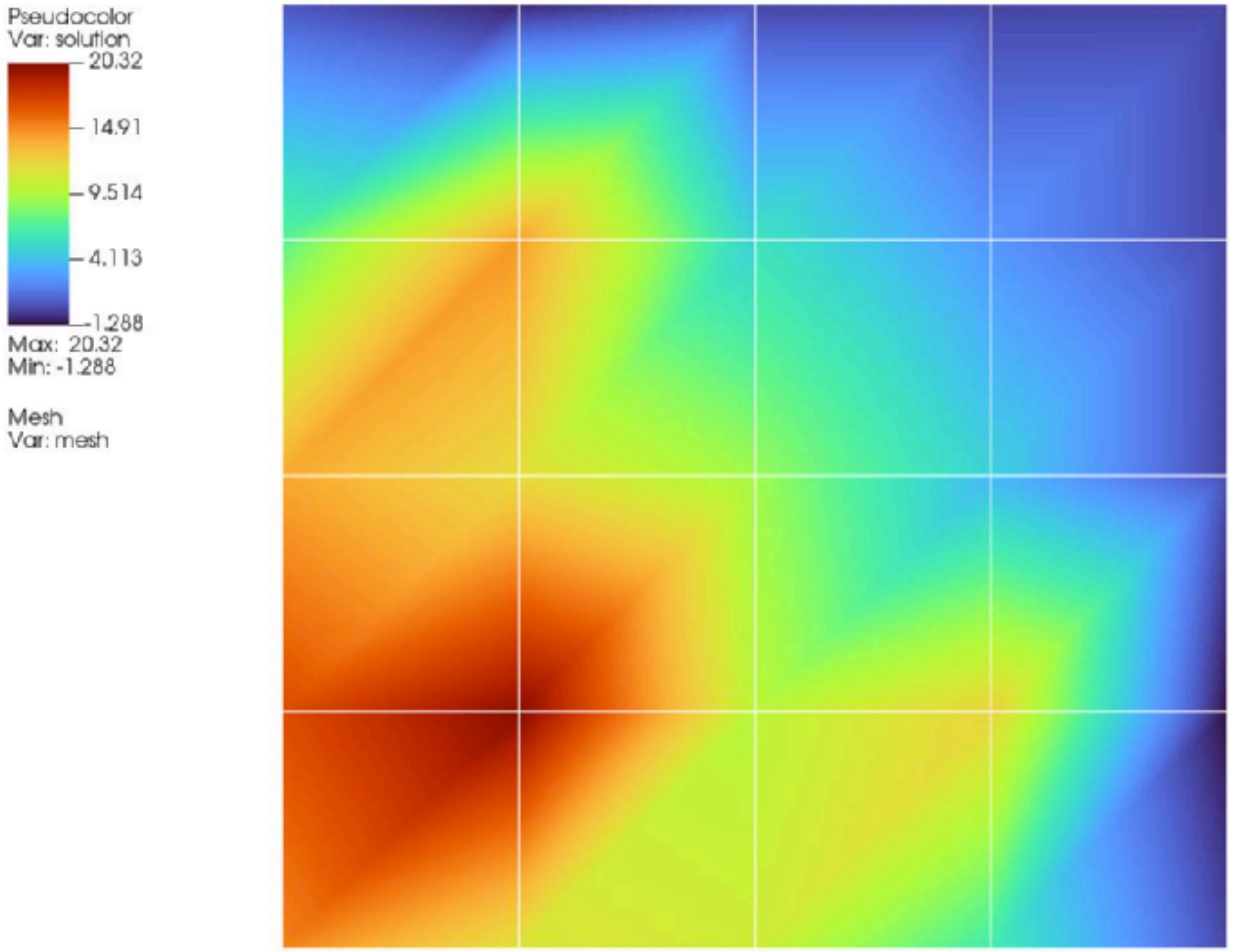
(difficult, heuristic) [1]

Mesh, Numerical solution



Adaptive mesh refinement strategies are difficult and heuristic

Idea: Use deep reinforcement learning to find a good strategy

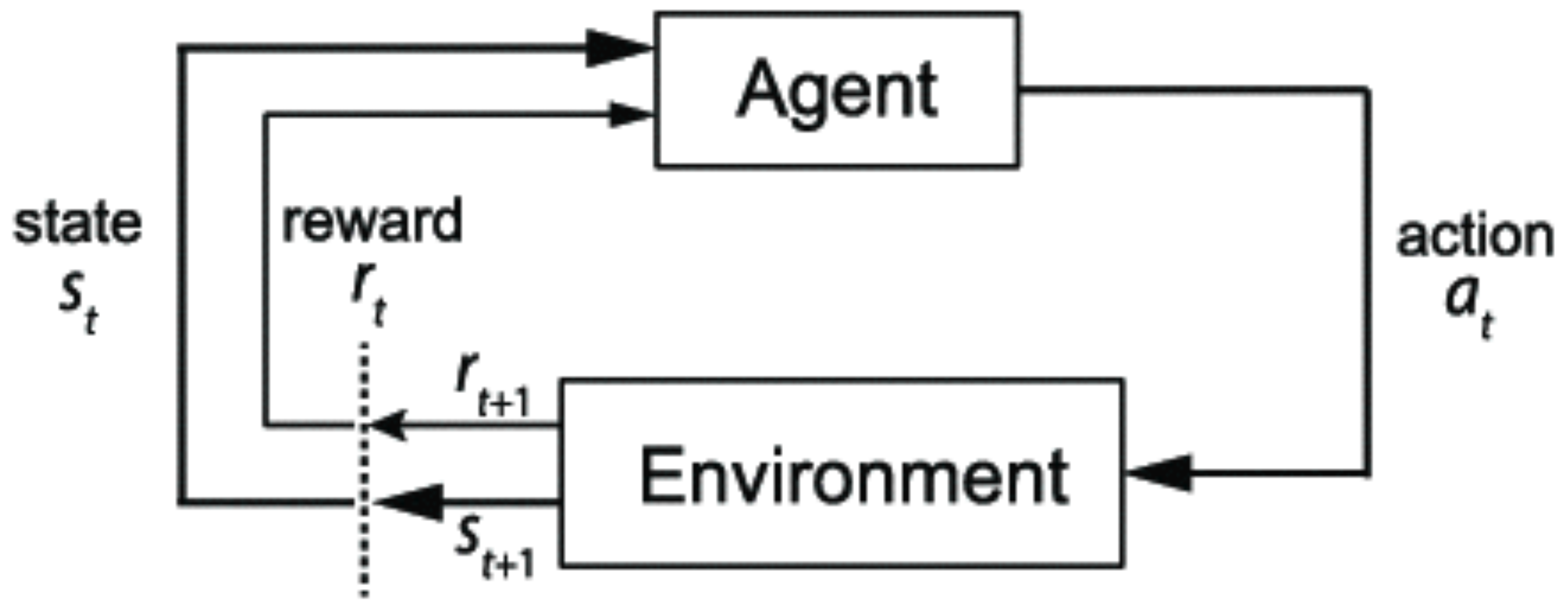


(Yang et al., [arxiv preprint] 2021)

Reinforcement learning

Agent: interacts with environment at discrete time intervals

- takes action a_t on state s_t
- receives reward r_{t+1}
- state transitions to s_{t+1}



Markov decision process: tuple (S, A, T, R, γ)

S - set of possible states, "state space"

A - set of possible actions, "action space"

$T(s', a, s) = \textbf{prob}\left(s_{t+1} = s' \mid s = s_t,\ a = a_t\right)$

R - set of possible rewards

$\gamma \in [0, 1]$ - time discount factor

Goal:

choose stochastic policy $\pi : S \to A$
that maximizes expected reward

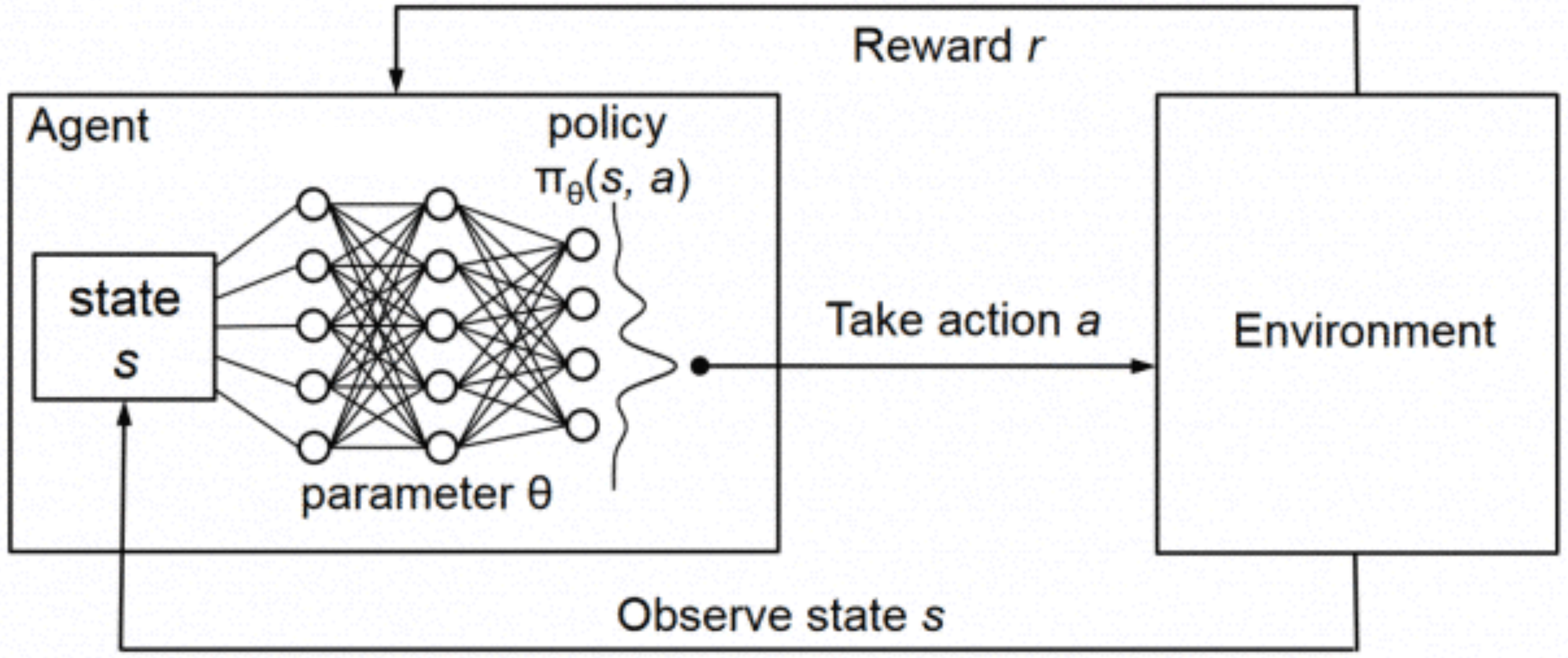
$$Q_t(s, a) = \mathbb{E}_\pi \left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \mid S_t = s,\ A_t = a \right]$$

over an infinite time horizon

Deep reinforcement learning

"Deep" RL:

- represent value function as neural network
- use each reward signal (experience) to update network during **training**
- once trained, agent is "**deployed**" by querying policy network for recommendation based on current state and action
- neural networks are universal function approximators [3] and can learn arbitrarily complex policies



goal:

choose stochastic policy $\pi : S \to A$
that maximizes expected reward

$$Q_t(s, a) = \mathbb{E}_\pi \left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \mid S_t = s, A_t = a \right]$$

over an infinite time horizon

How can we make our numerical solver "play against itself?"

Original problem:

$$\nabla \cdot (\mathbf{c}u) = f$$

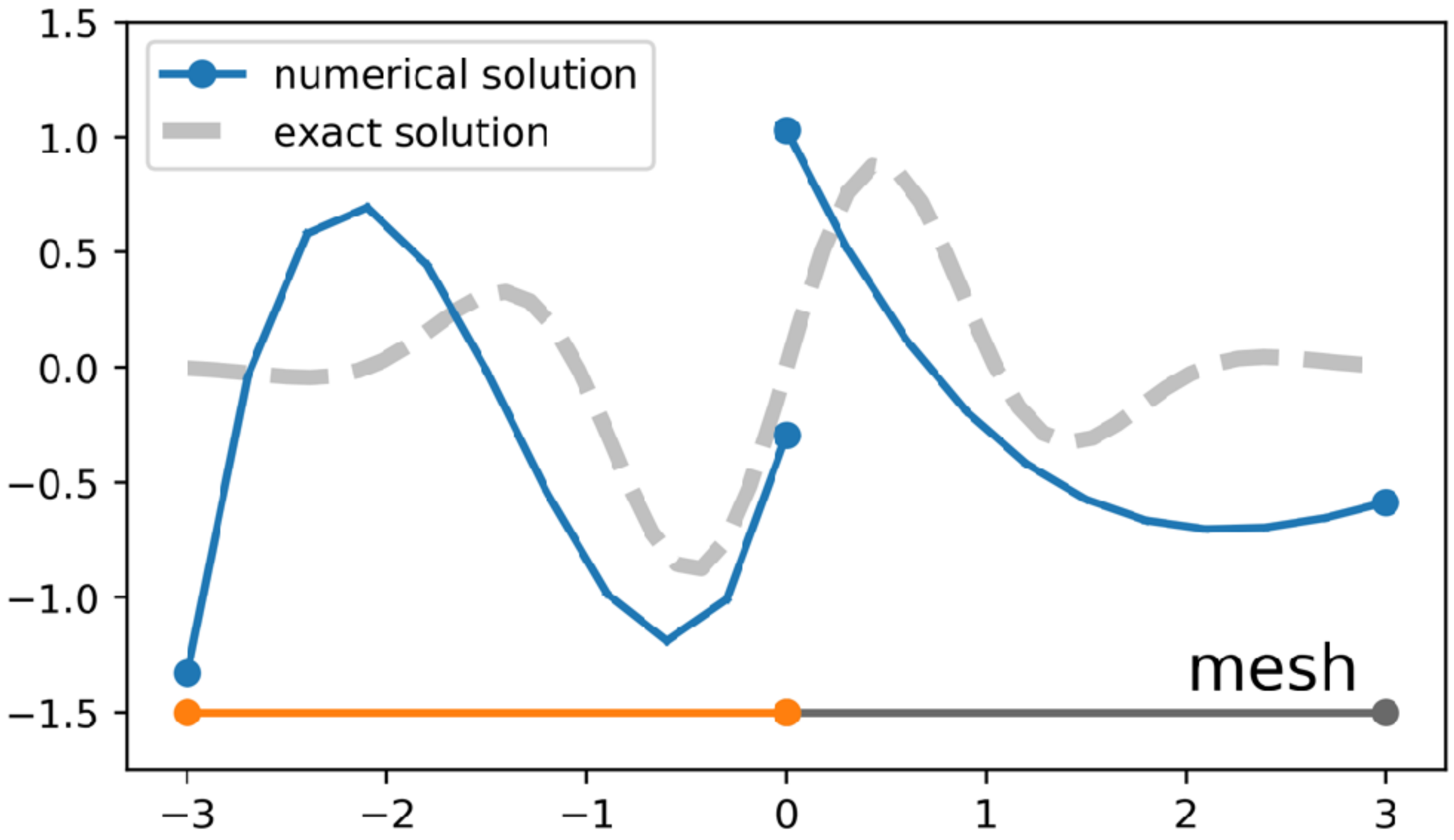
$$u = g_D \qquad \text{on } \Gamma_{\text{inflow}}$$

Weak formulation [1]**:**

find u_h such that

$$-(\nabla w, \mathbf{c}u_h)_\Omega + \langle w, \hat{u}_h(\mathbf{c}\cdot\mathbf{n})\rangle_{\partial\Omega} = (w, f)_\Omega$$

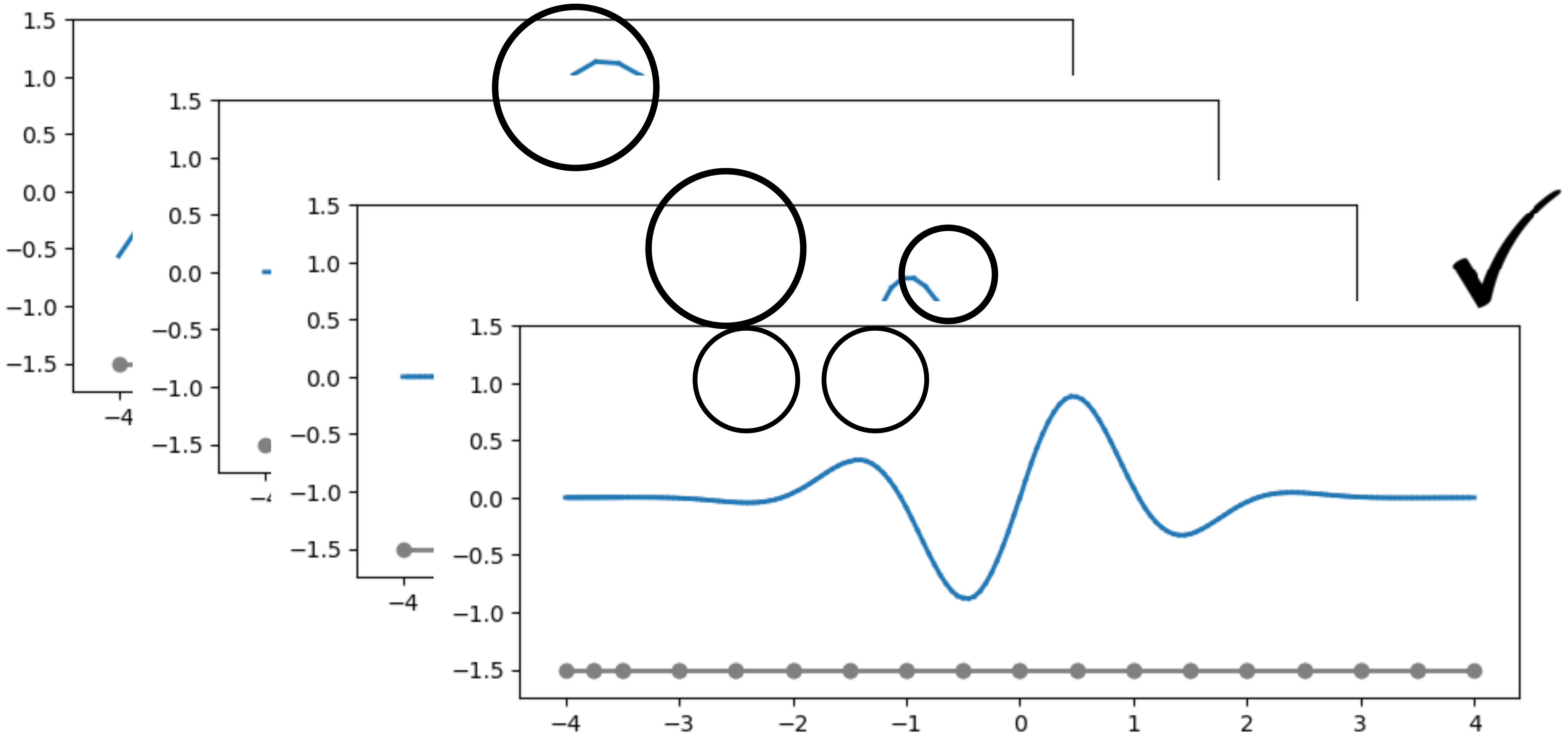
for all $w \in \mathcal{P}^p$, the space of discontinuous polynomials of order p on each element

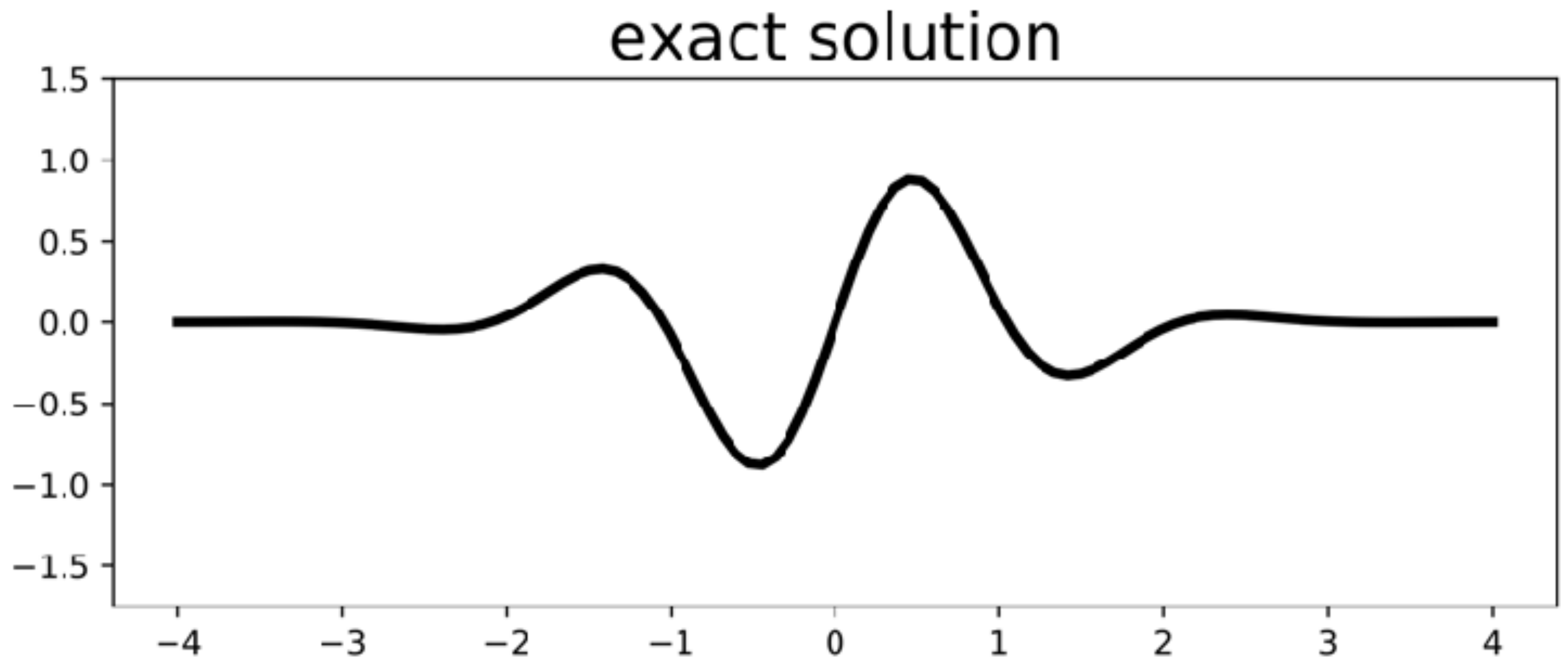


Increasing conformity upon refinement

Key idea:

- underlying solution to PDE is **continuous**
- numerical representation of solution is **discontinuous**

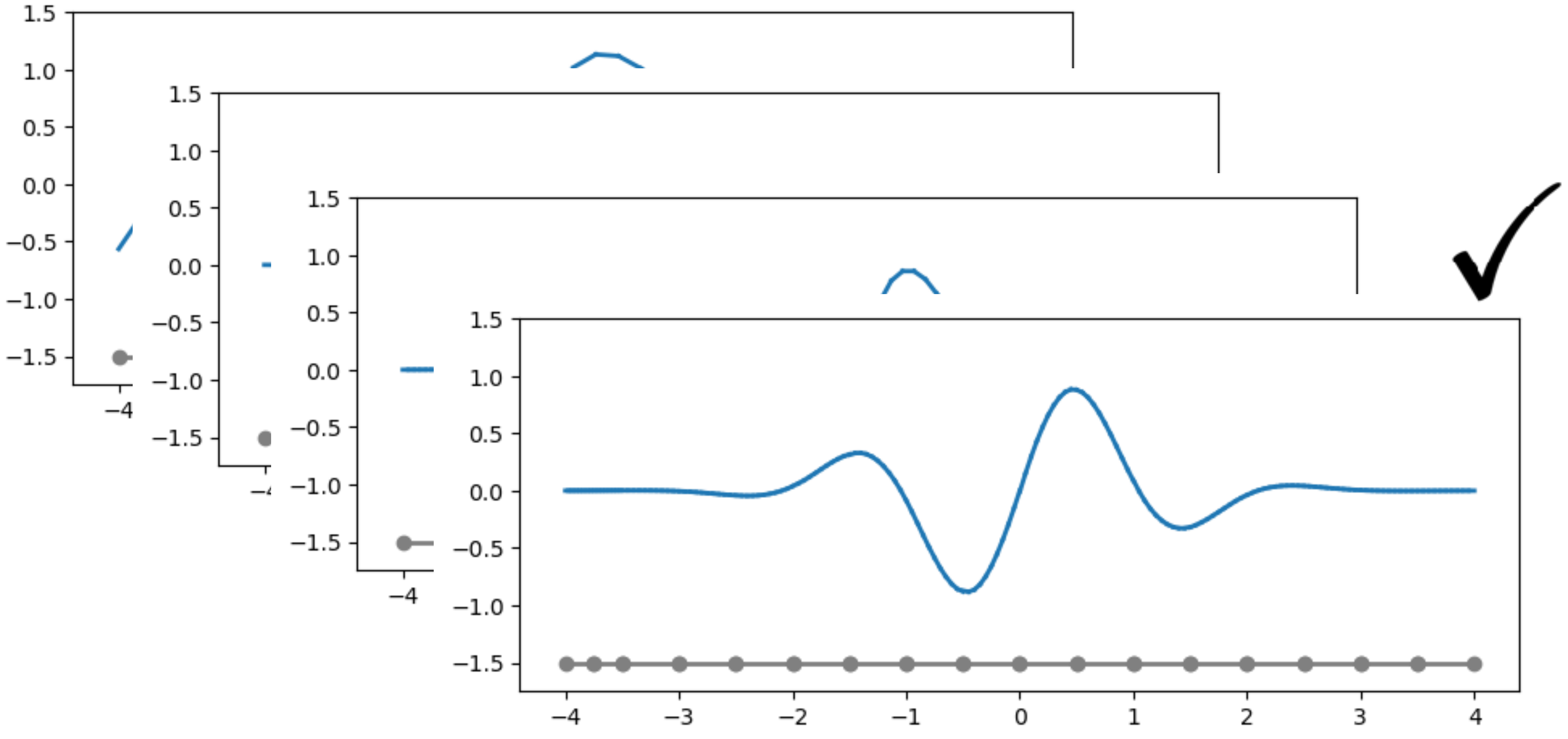


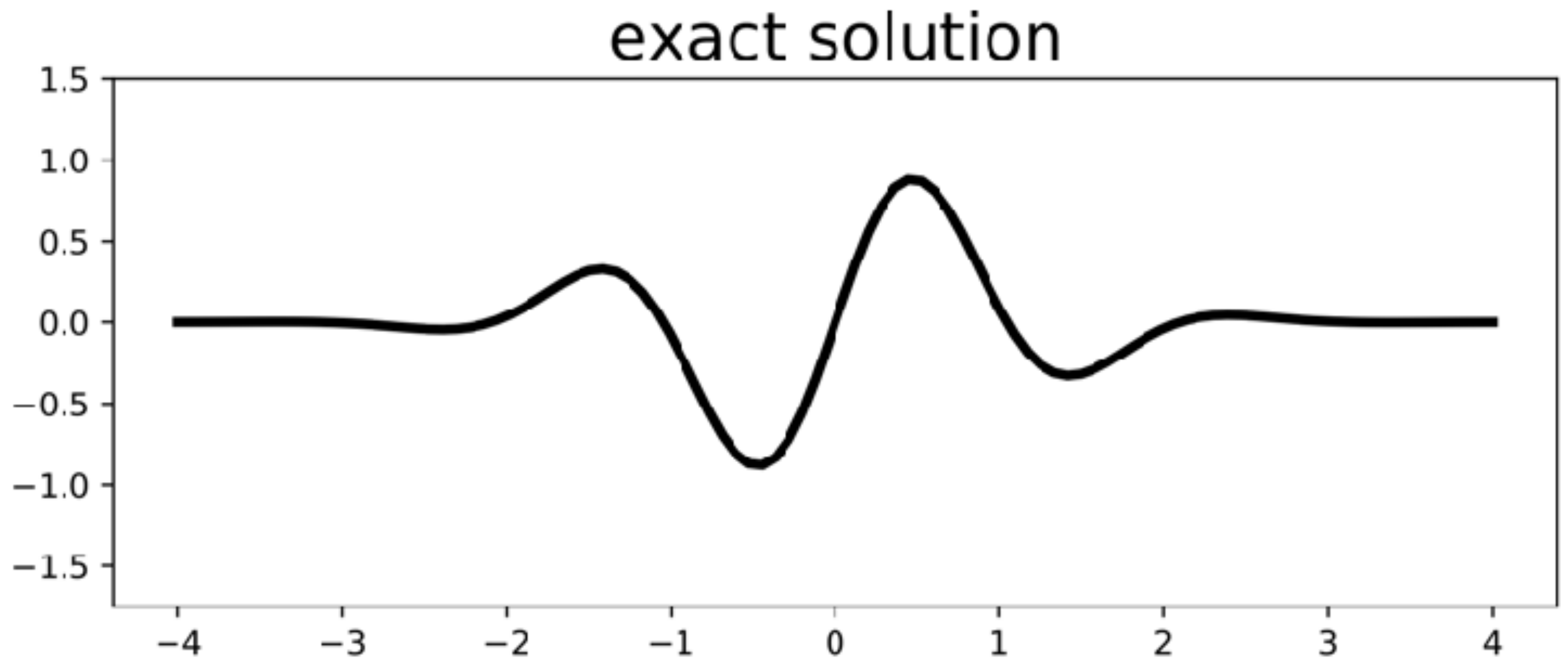


Increasing conformity upon refinement

Key idea:

- underlying solution to PDE is **continuous**
- numerical representation of solution is **discontinuous**

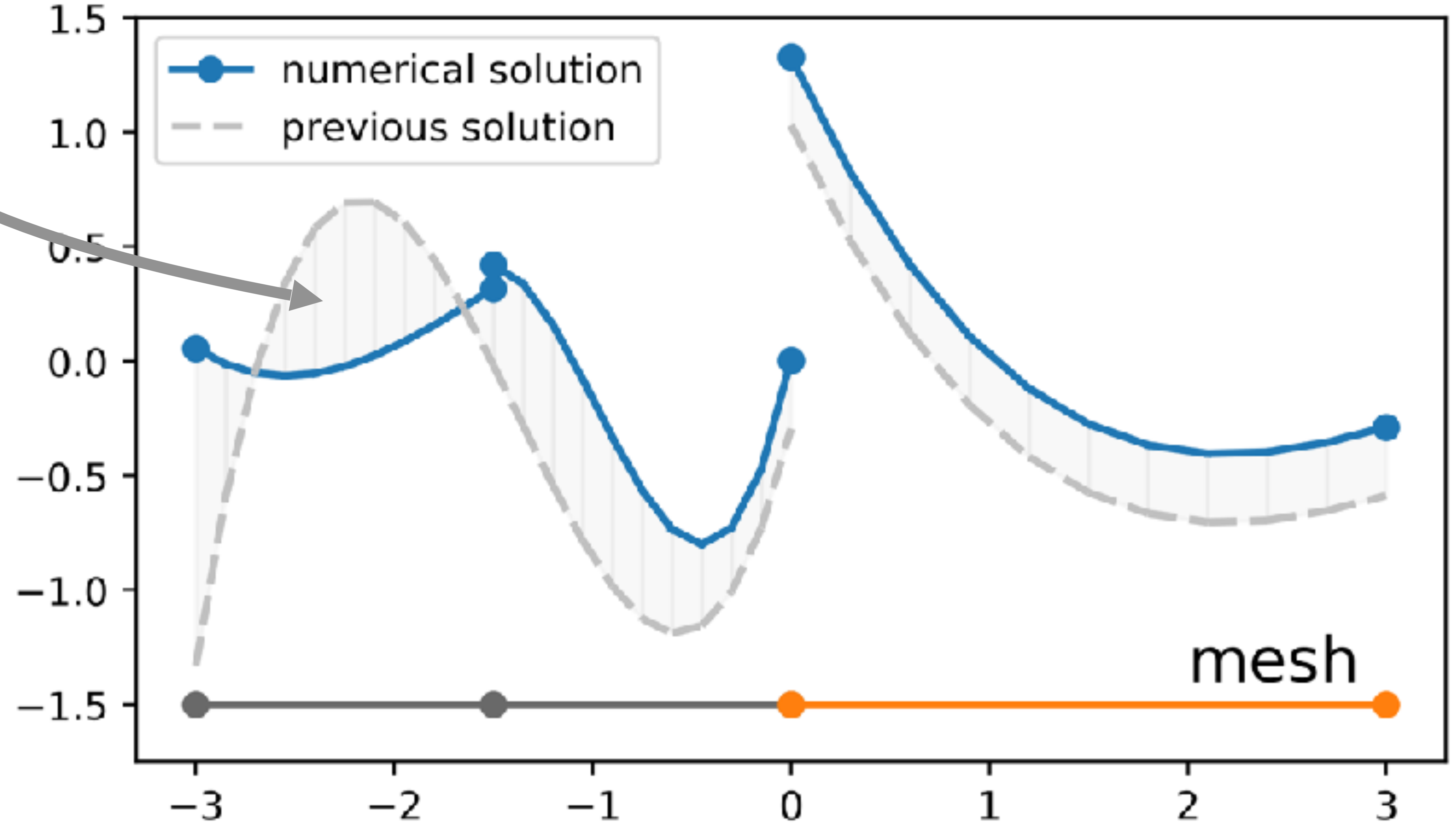




Designing the reward function

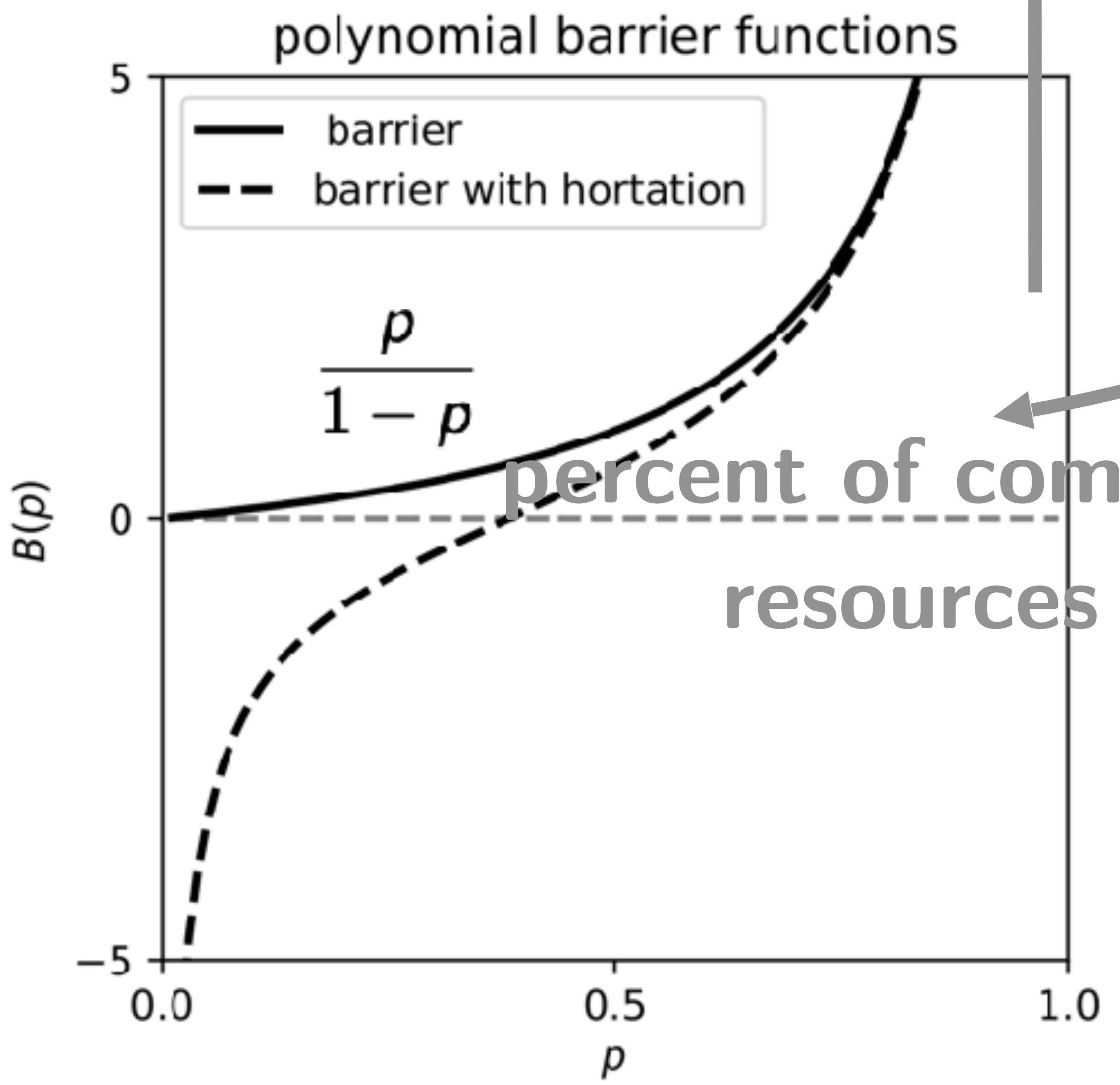
"Change in u_h as a result of refinement"

$$+\ \underline{\Delta u_h} = \sum_{K \in \mathcal{T}_h} \int_K |u_h^{t+1} - u_h^t| \, dK$$



"Computational cost incurred"

$$-\ \gamma_c \cdot [\text{cost}] \cdot \underline{B(p)}$$

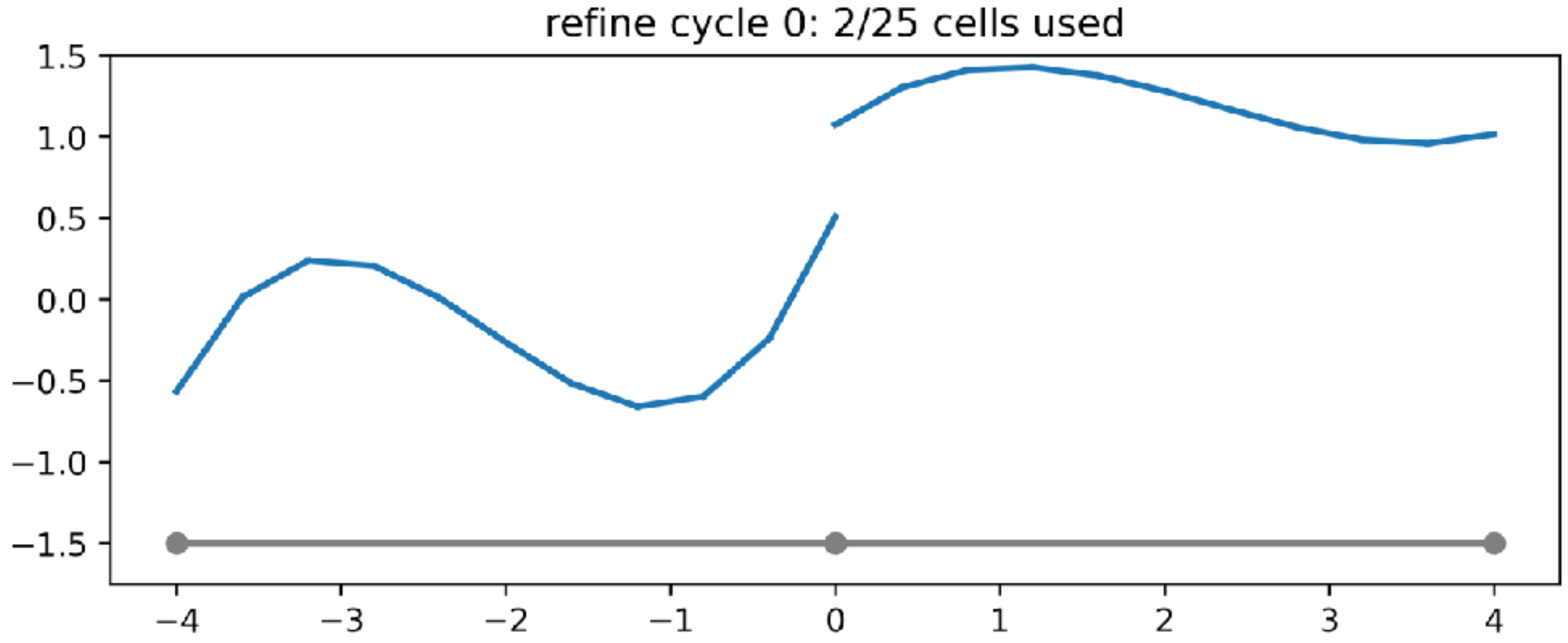


percent of computational resources used

Training

Single RL agent

Current numerical solution:

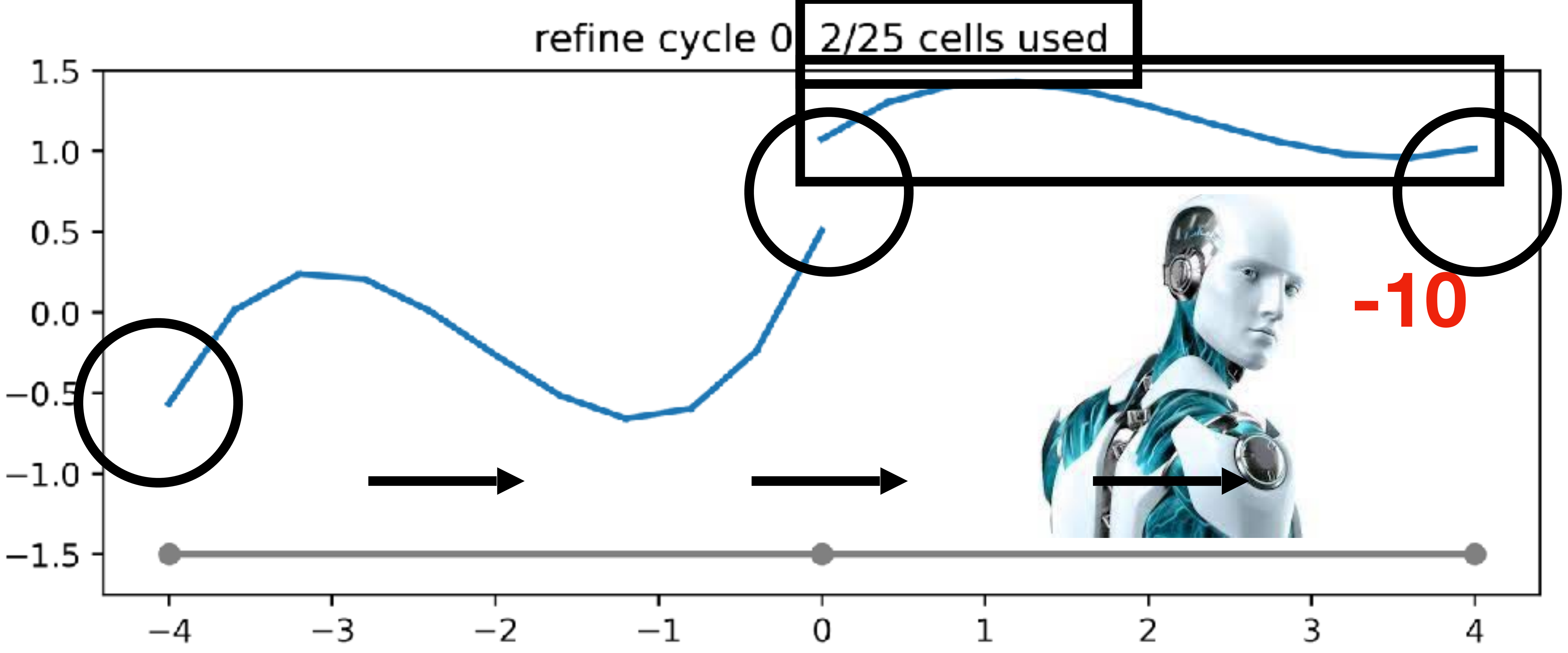


Training

Observation space:

- the local solution
- the interface jumps over the cell boundary
- the average interface jump over all mesh interfaces
- the current usage of computational resources
- any PDE-specific features the user wishes to provide

Current numerical solution:



Action space:

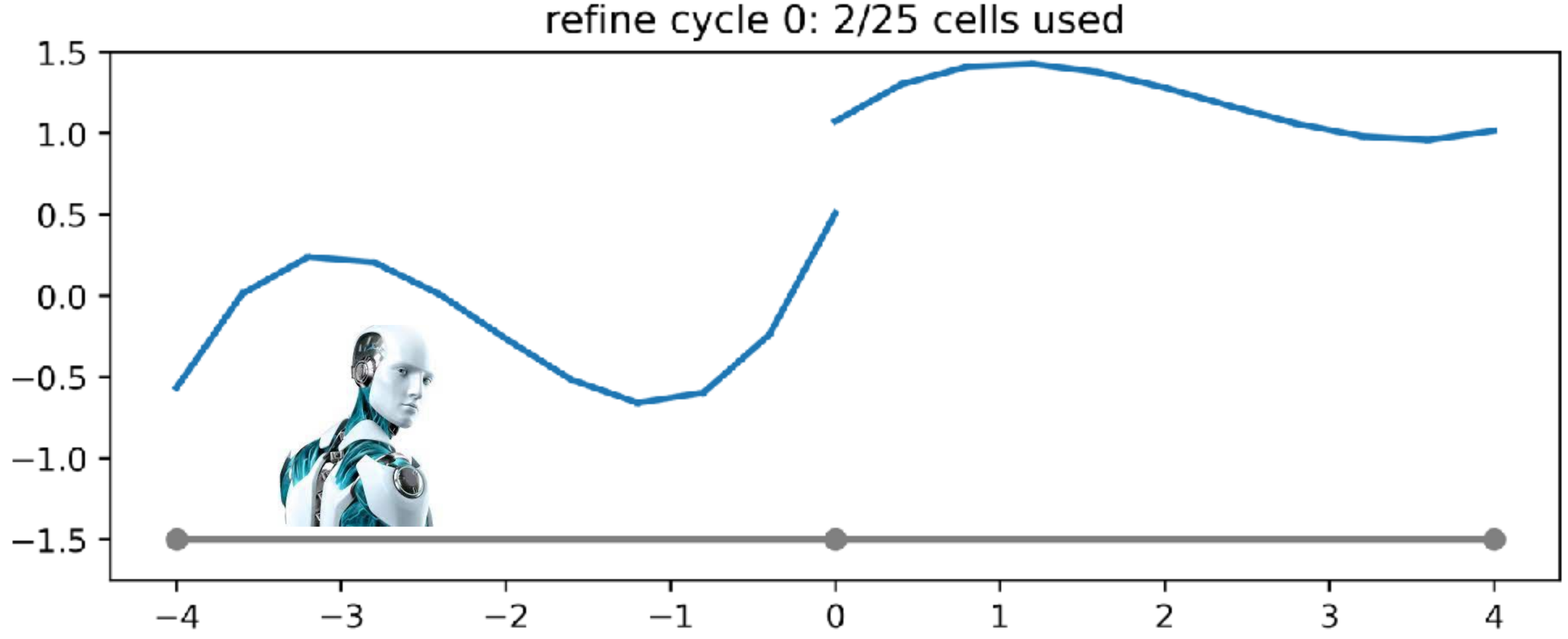
- refine
- do nothing
- coarsen

Training

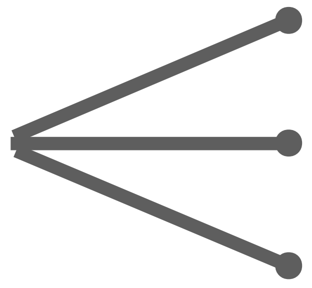
Observation space:

- the local solution
- the interface jumps over the cell boundary
- the average interface jump over all mesh interfaces
- the current usage of computational resources
- whatever PDE-specific features you want to provide

Current numerical solution:



Action space:



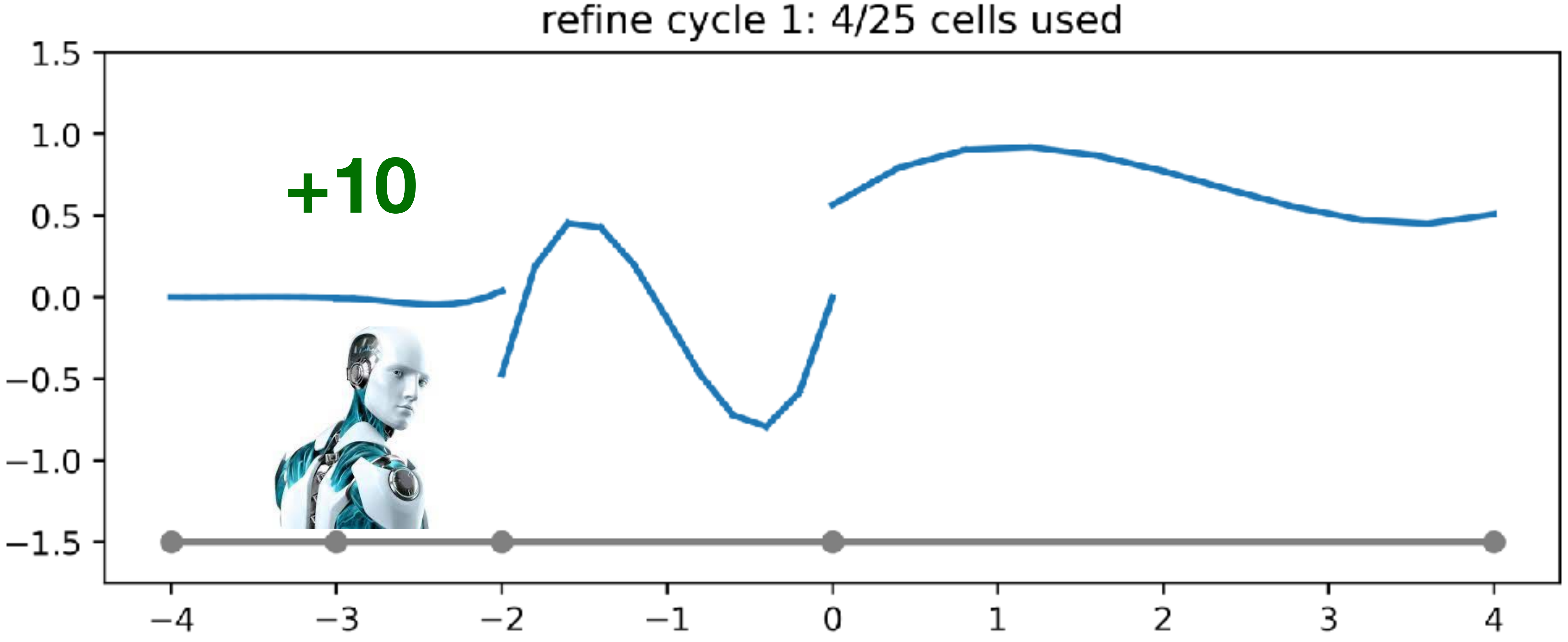
refine
do nothing
coarsen

Training

Observation space:

- the local solution
- the interface jumps over the cell boundary
- the average interface jump over all mesh interfaces
- the current usage of computational resources
- whatever PDE-specific features you want to provide

Current numerical solution:

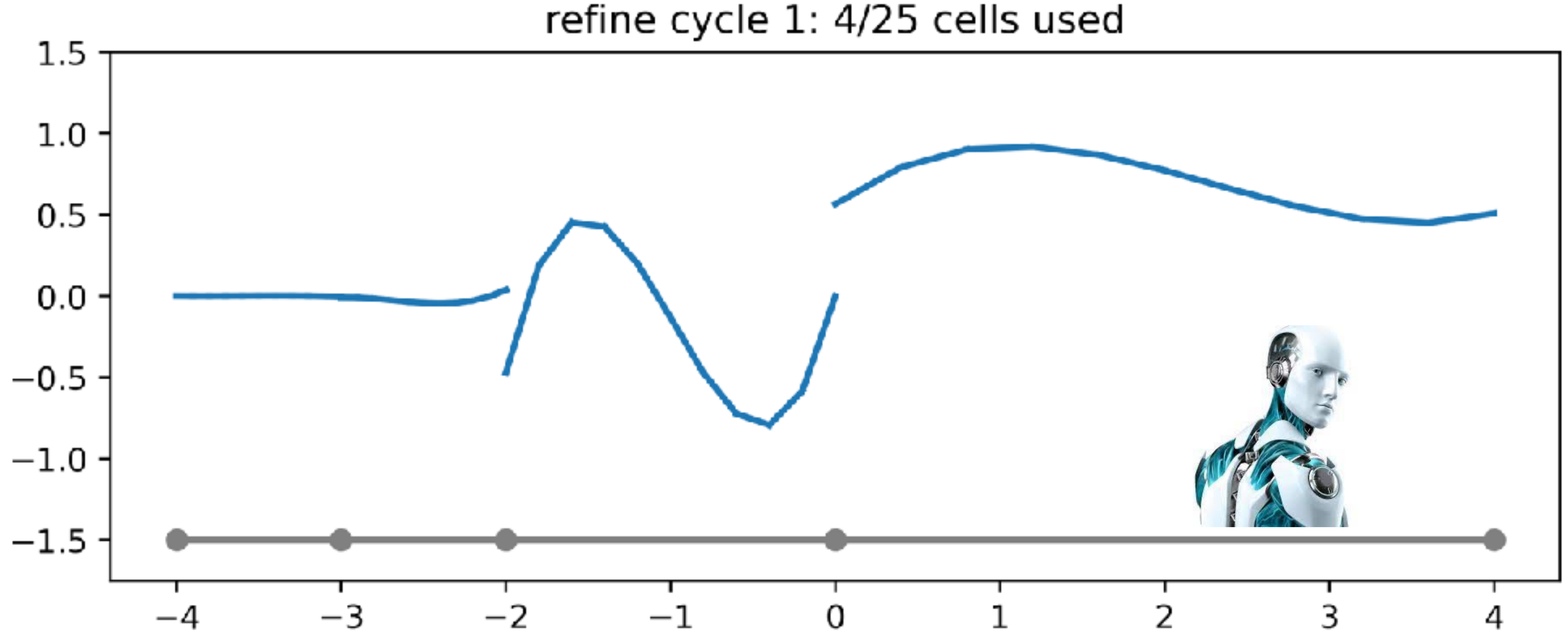


Action space:

- refine
- do nothing
- coarsen

Training

Current numerical solution:



Observation space:

- the local solution
- the interface jumps over the cell boundary
- the average interface jump over all mesh interfaces
- the current usage of computational resources
- whatever PDE-specific features you want to provide

Action space:

- refine
- do nothing
- coarsen

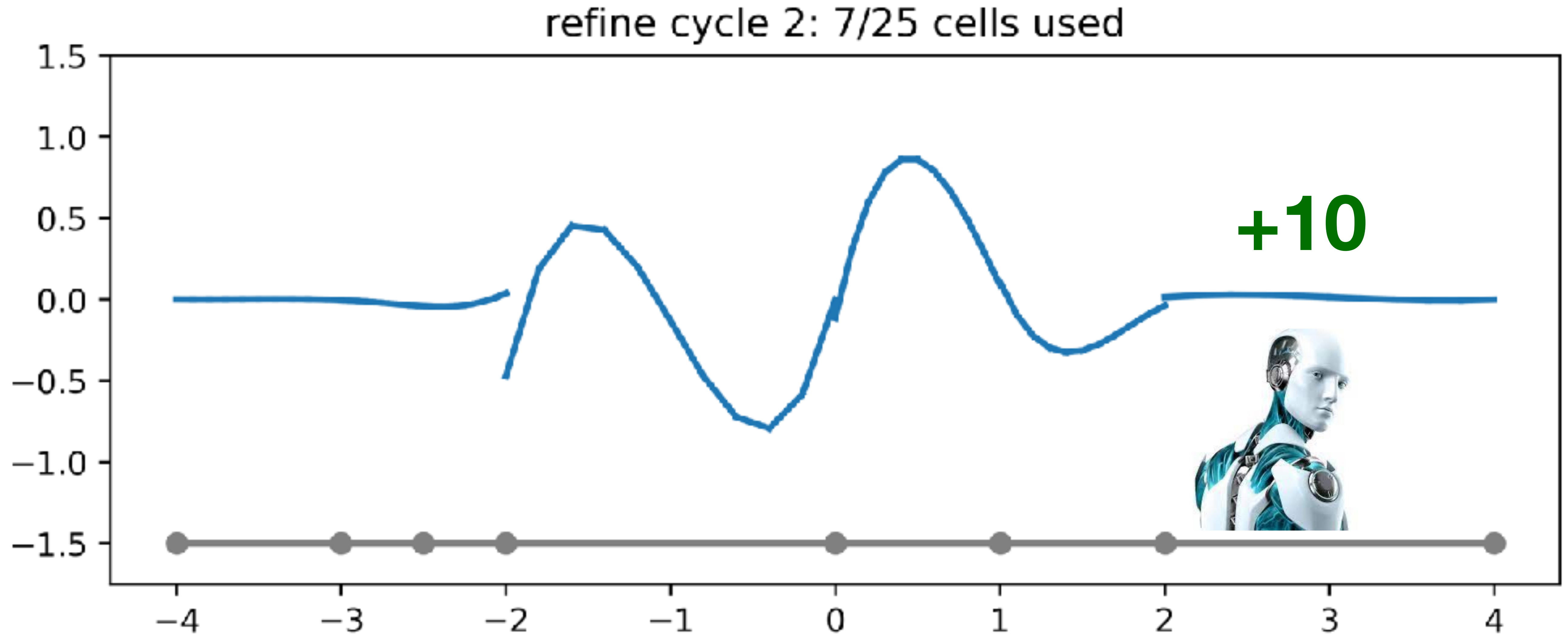
[1] Foucart et al., JCP. 2022.

Training

Observation space:

- the local solution
- the interface jumps over the cell boundary
- the average interface jump over all mesh interfaces
- the current usage of computational resources
- whatever PDE-specific features you want to provide

Current numerical solution:



Action space:

- refine
- do nothing
- coarsen

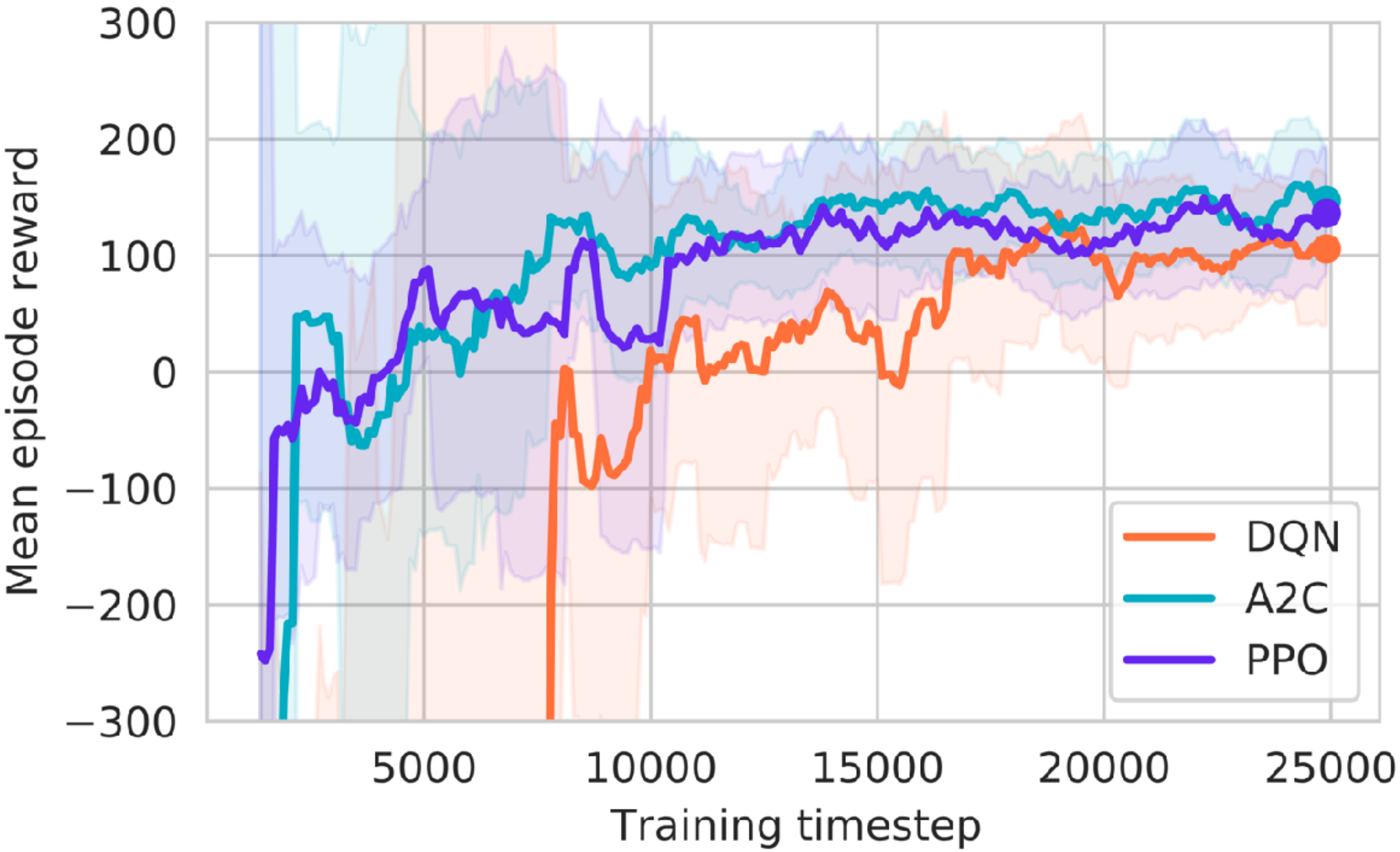
Training

This is the numerical solver "playing against itself"

Over time, it learns a good refinement strategy.

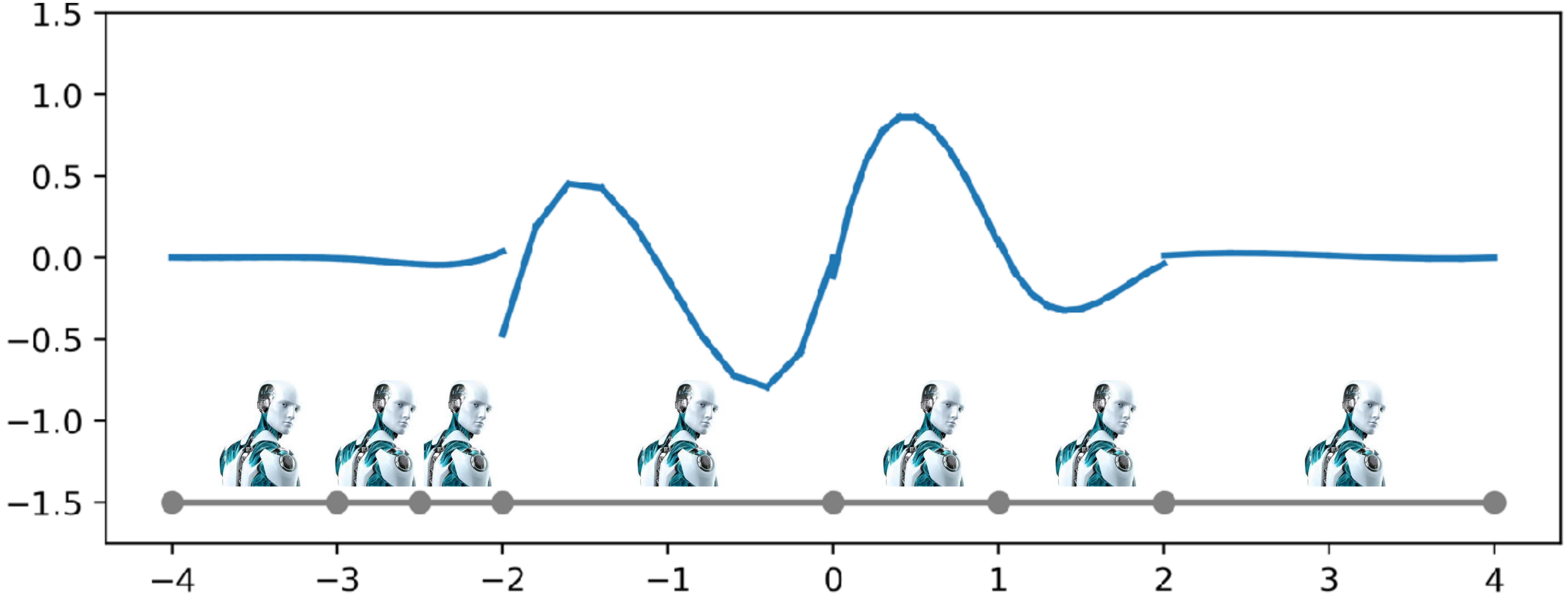
Observation space:

- the local solution
- the interface jumps over th boundary
- the average interface jump all mesh interfaces
- the current usage of computational resources
- whatever PDE-specific feat you want to provide



Deployment:

Walk every cell with the RL-agent



and refine according to its recommendation.

Proof of concept: smooth jump test case

Steady-state advection equation

$$\nabla \cdot (\mathbf{c}u) = f$$

$$u = g_D \qquad \text{on } \Gamma_{\text{inflow}}$$

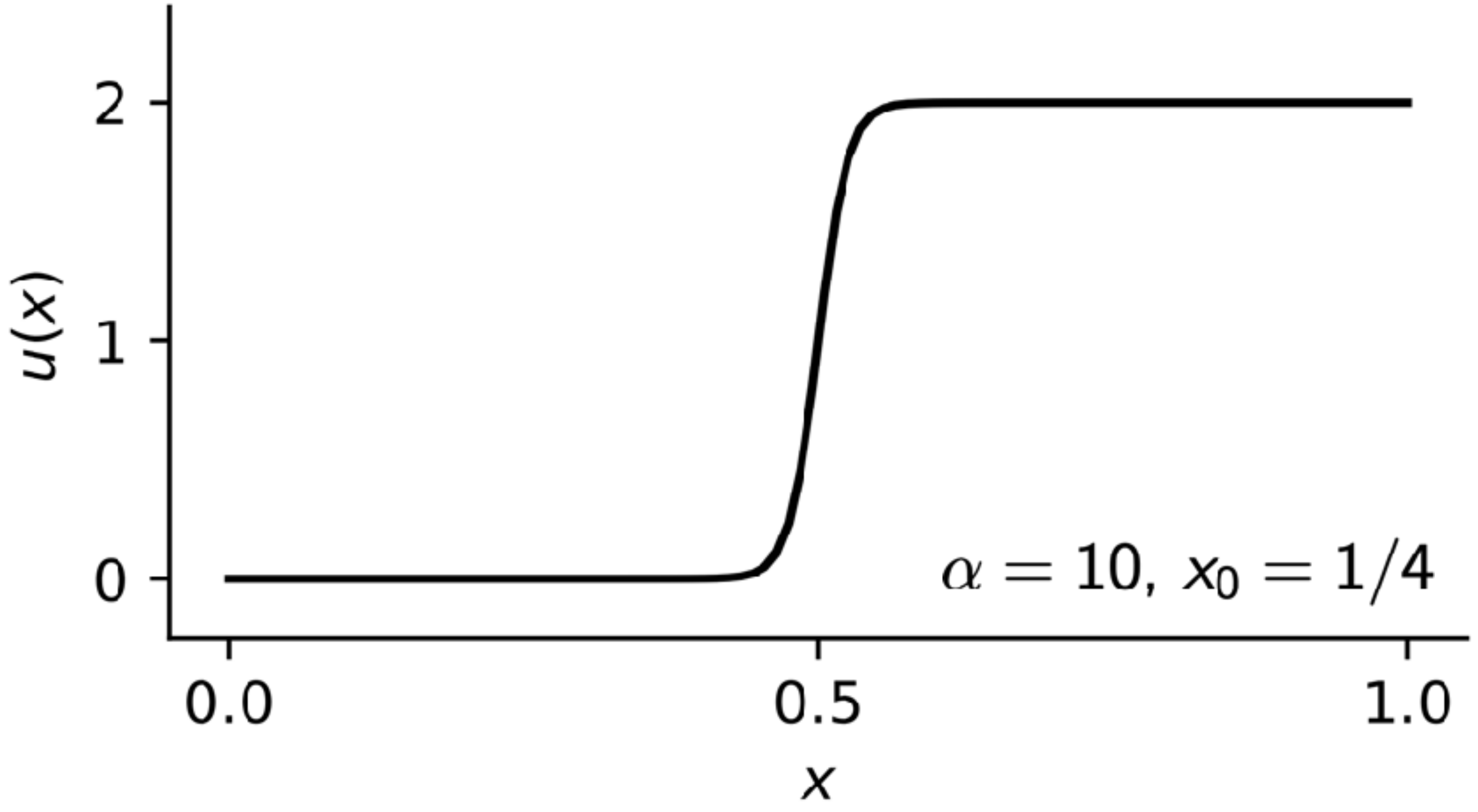
velocity $\mathbf{c} = 1$

$\Gamma_{\text{inflow}} = \{x \in \Gamma : \mathbf{c} \cdot \mathbf{n} \leq 0\}$

***at no point** during training or deployment do we make use of the exact solution

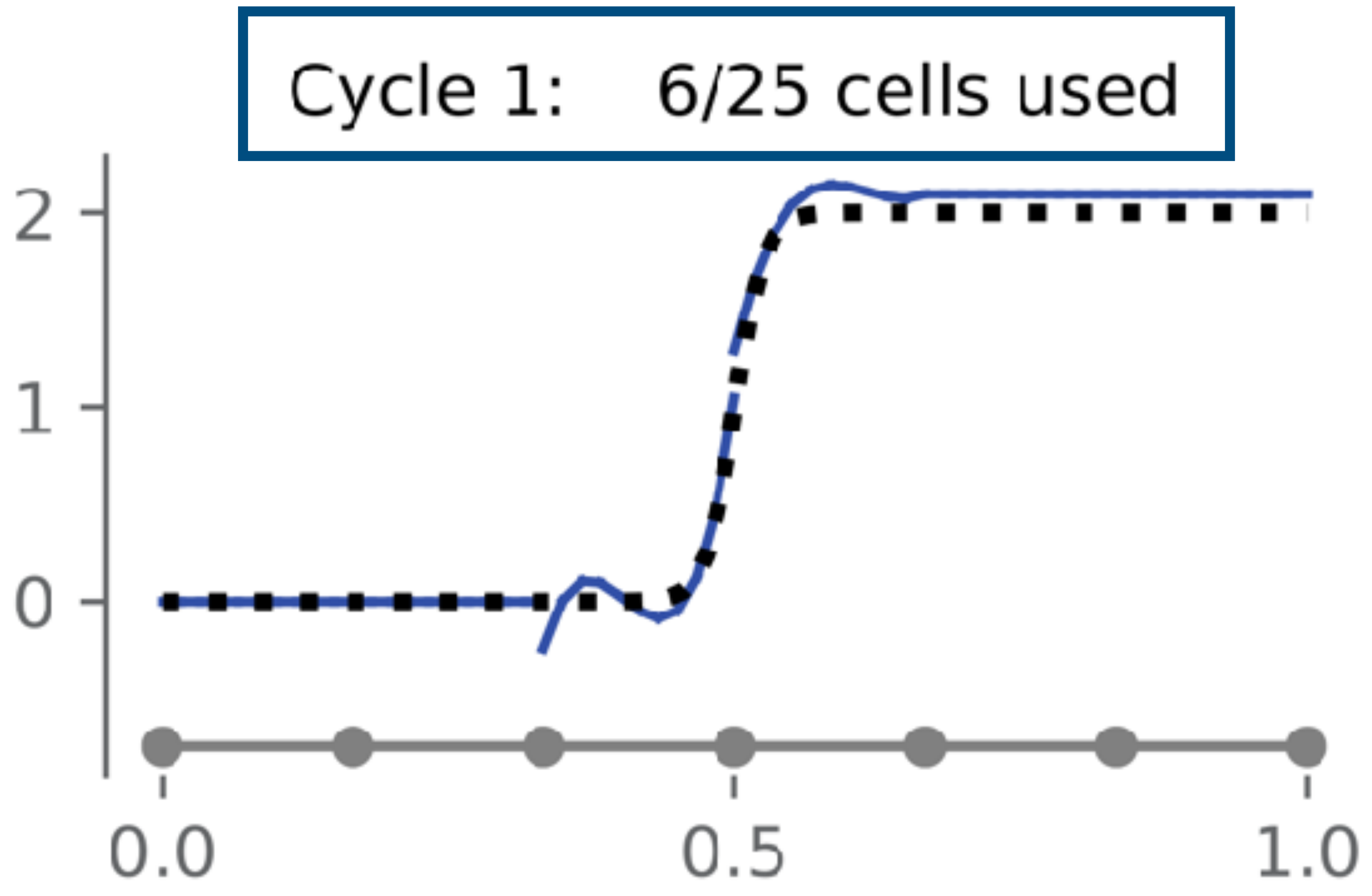
Exact solution*

$$u(x) = 1 - \tanh[\alpha(1 - 4(x - x_0))]$$



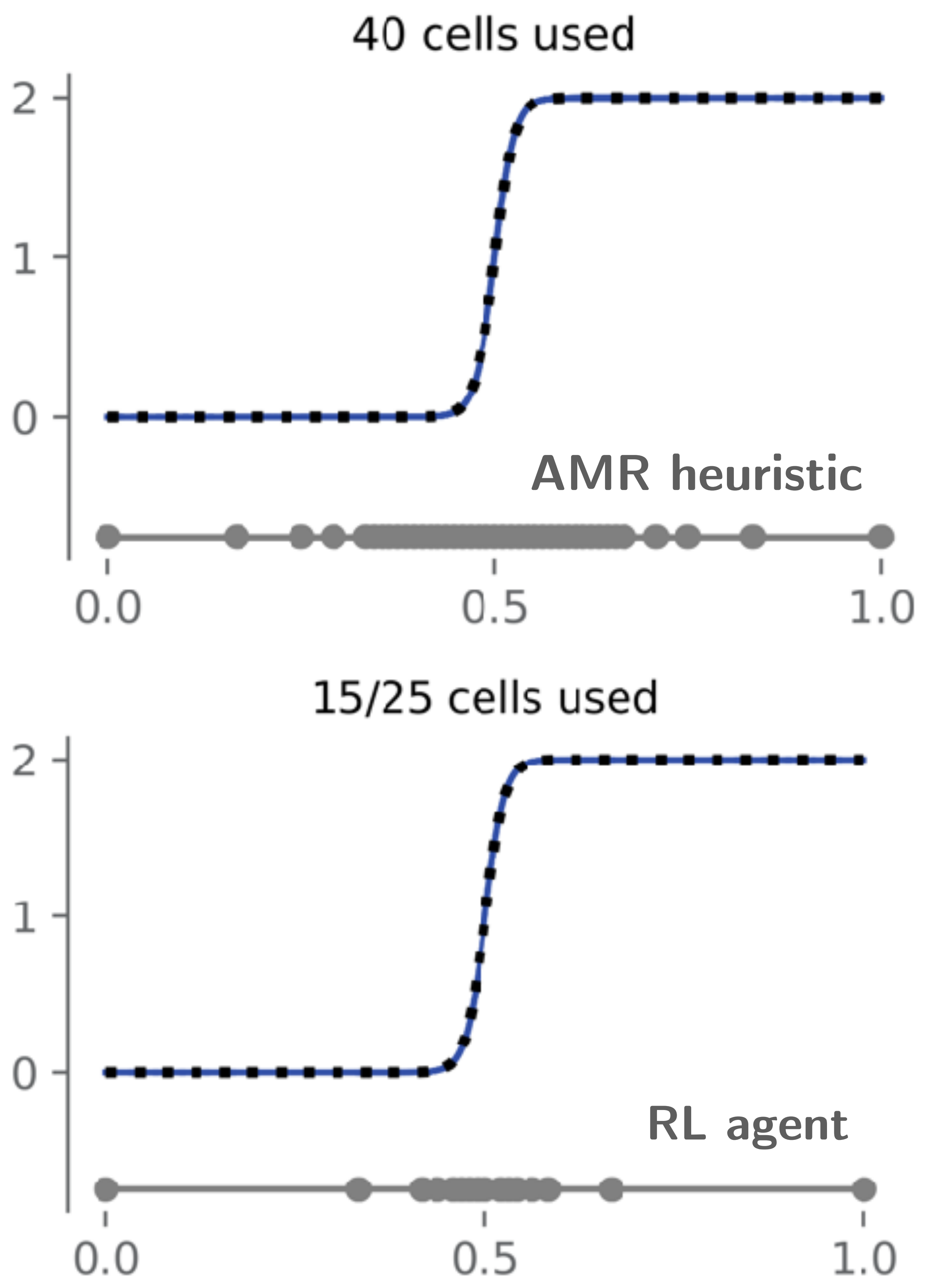
Trained model deployment

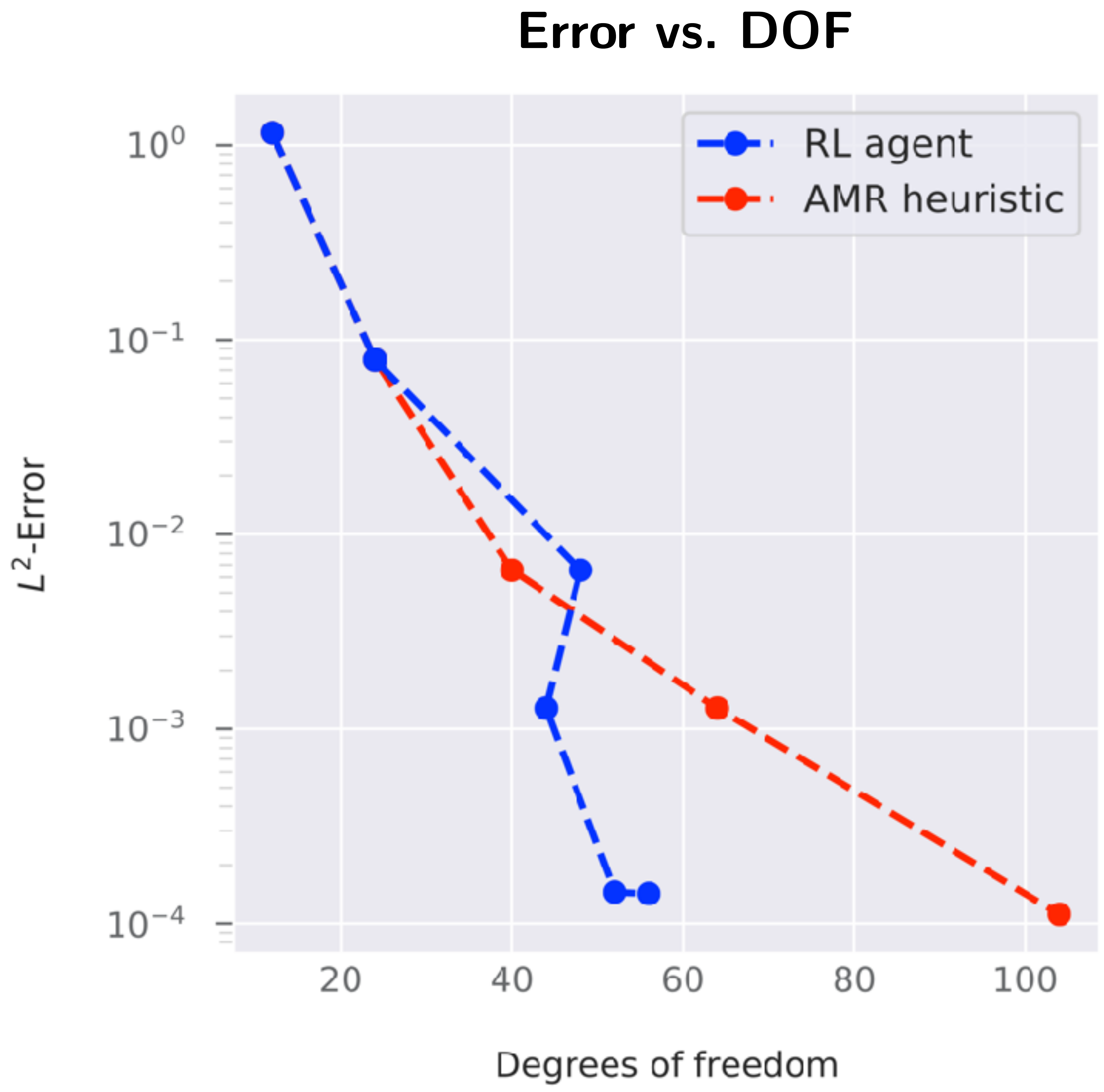
RL Agent, 25 cell budget



Test against a typical AMR implementation

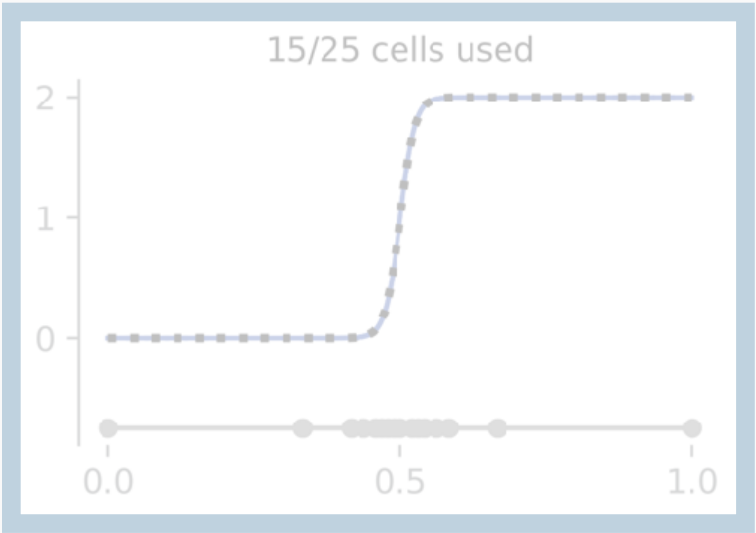
(gradient-based, 50/50 bulk-refinement)



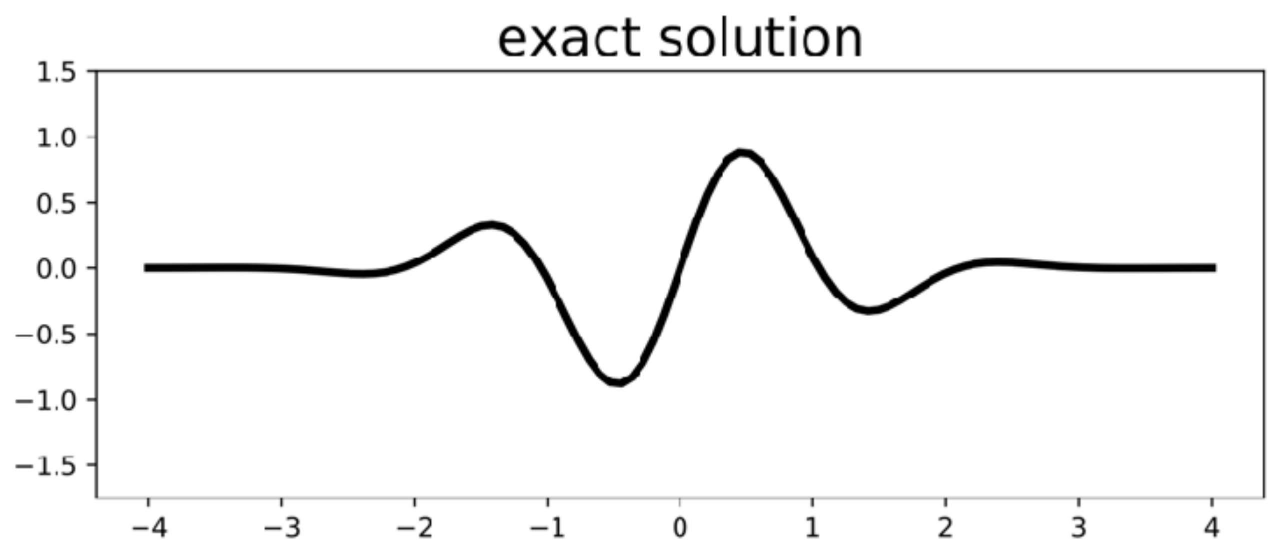


Does it generalize?

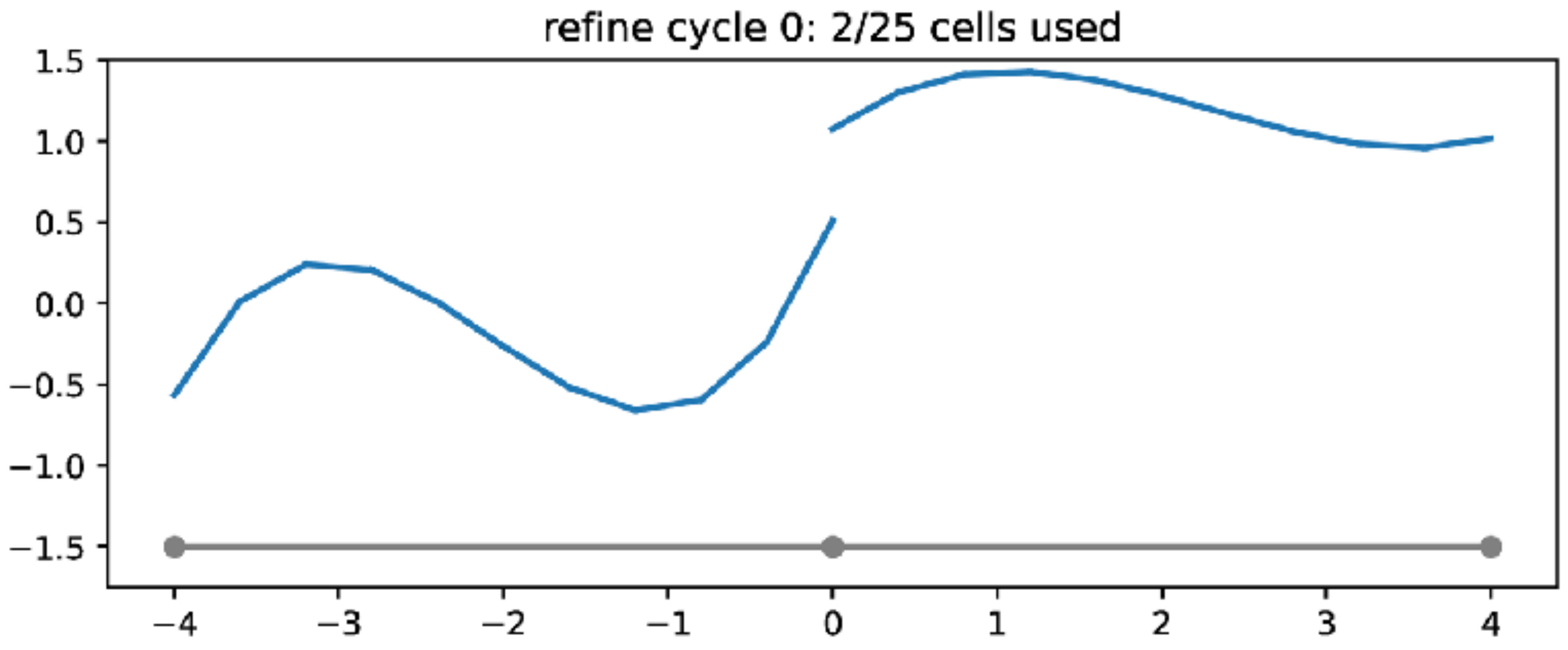
Same trained RL model as in previous example



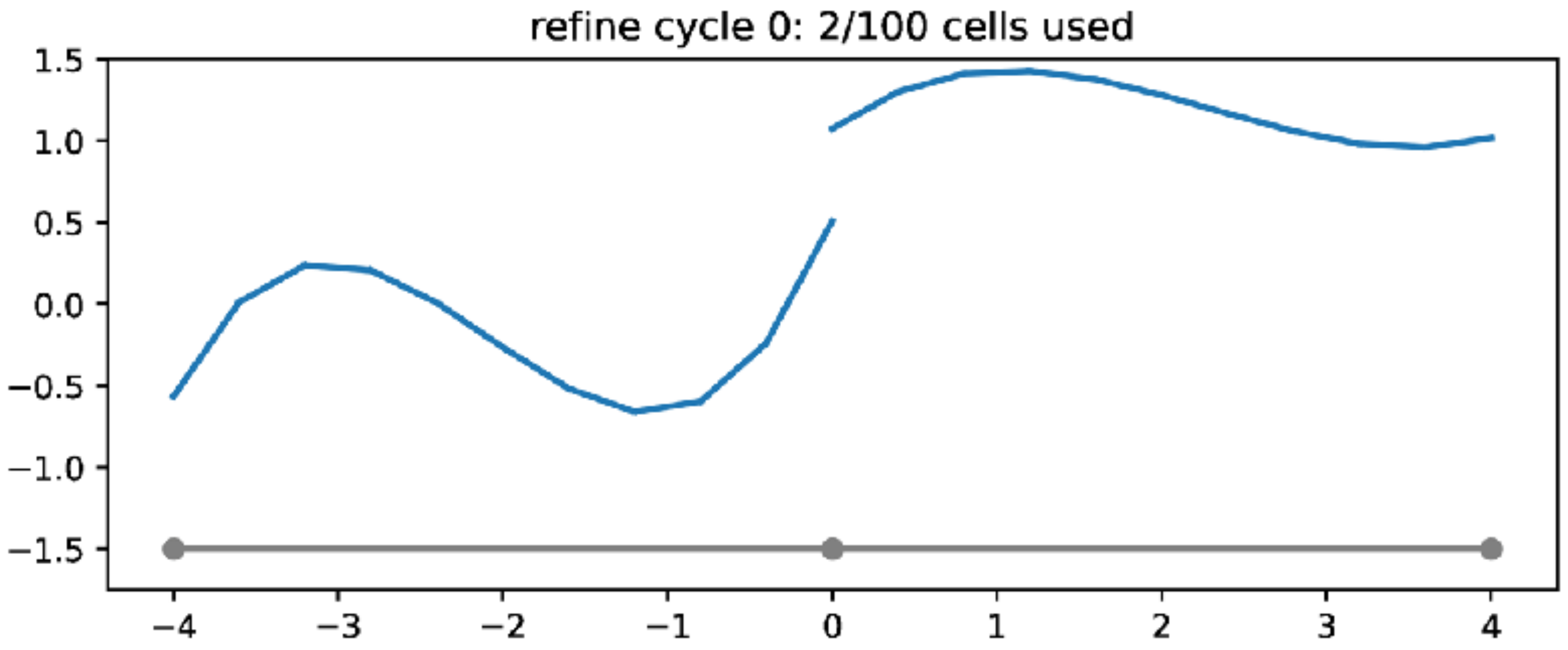
- **No additional training**
- **Apply it to same PDE, but with new boundary conditions and forcing function**
- **Does it still recommend a good mesh?**



RL Agent, 25 cell budget



RL Agent, 100 cell budget



Time-dependent problems

Sommerfeld wave equation

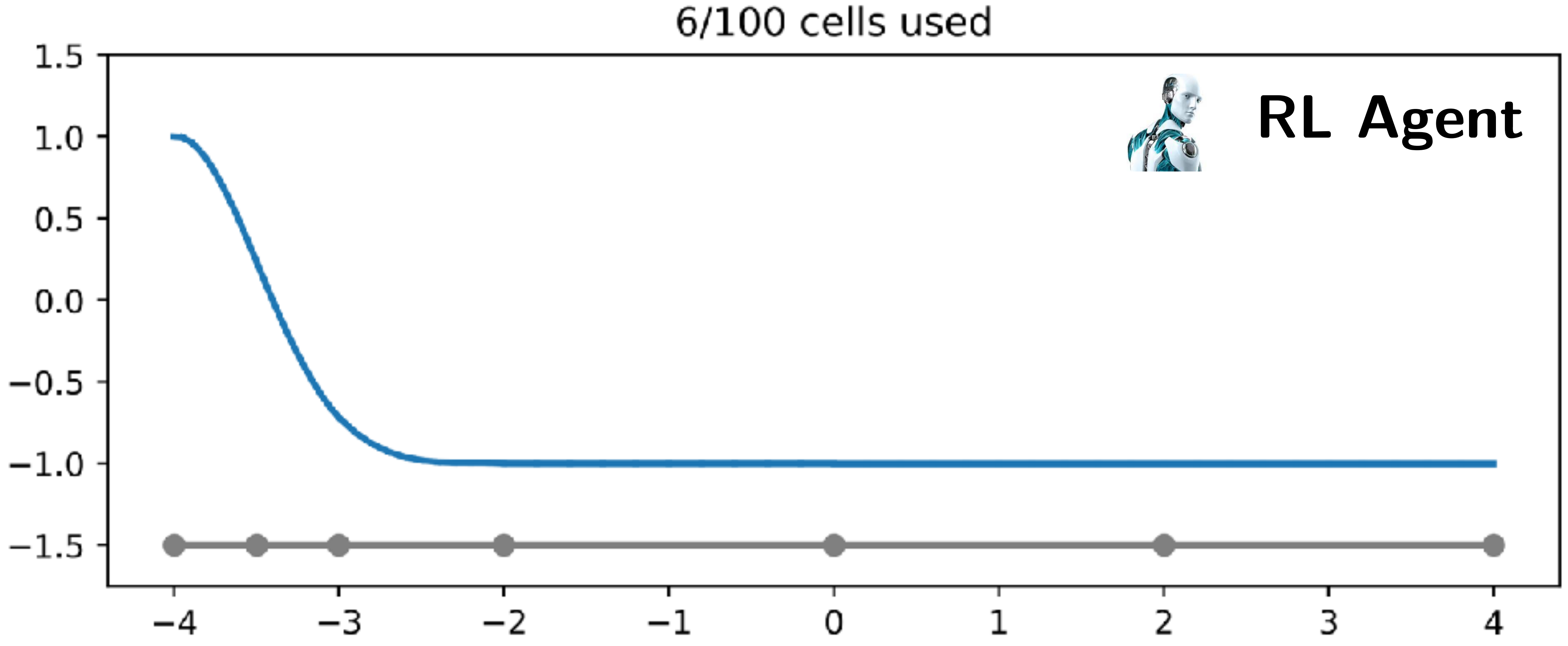
$$\frac{\partial u}{\partial t} + \frac{\partial u}{\partial x} = 0$$

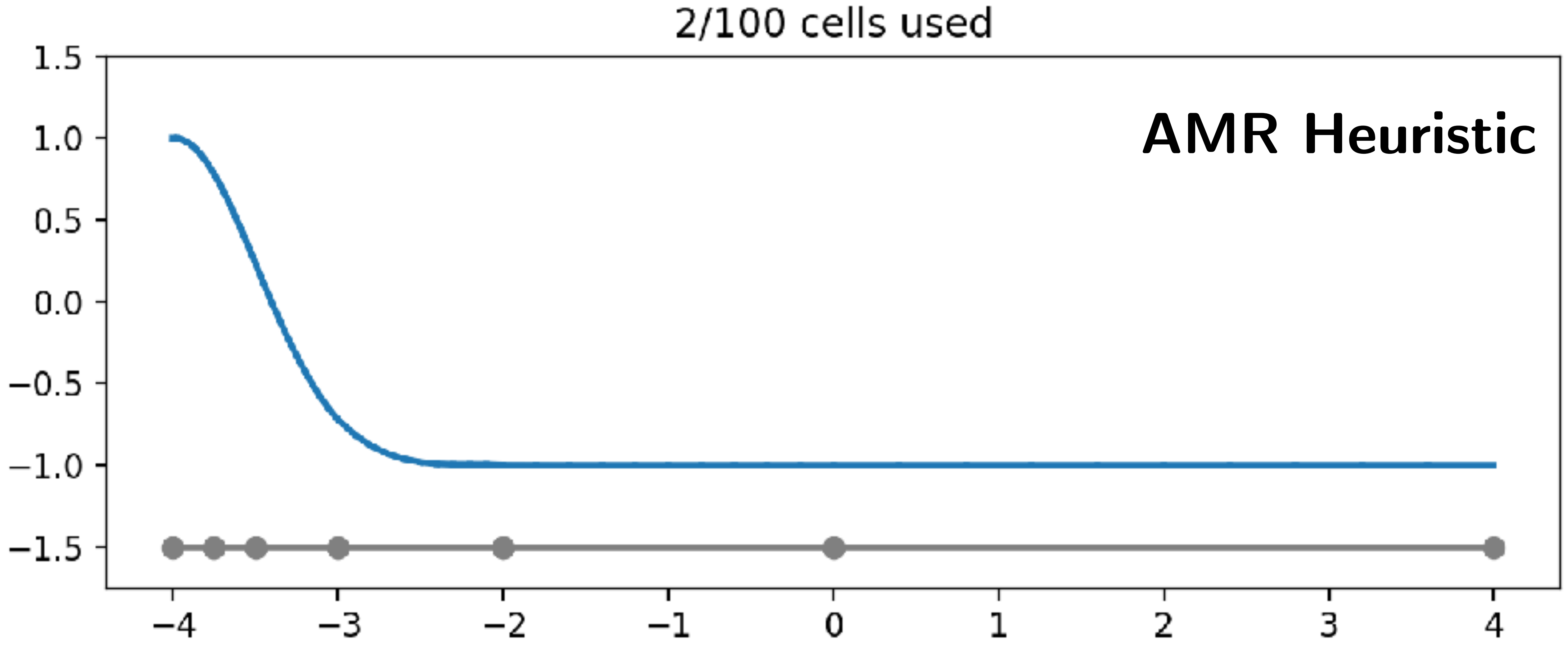
initial condition:

$$u_0 = \exp\left(\frac{1}{2\sigma^2}(x-\mu)^2\right)$$

$$\sigma^2 = 0.25 \qquad \mu = -4$$

RL agent strategy preserves solution features, at a fraction of the cost.





Higher-dimensional problems

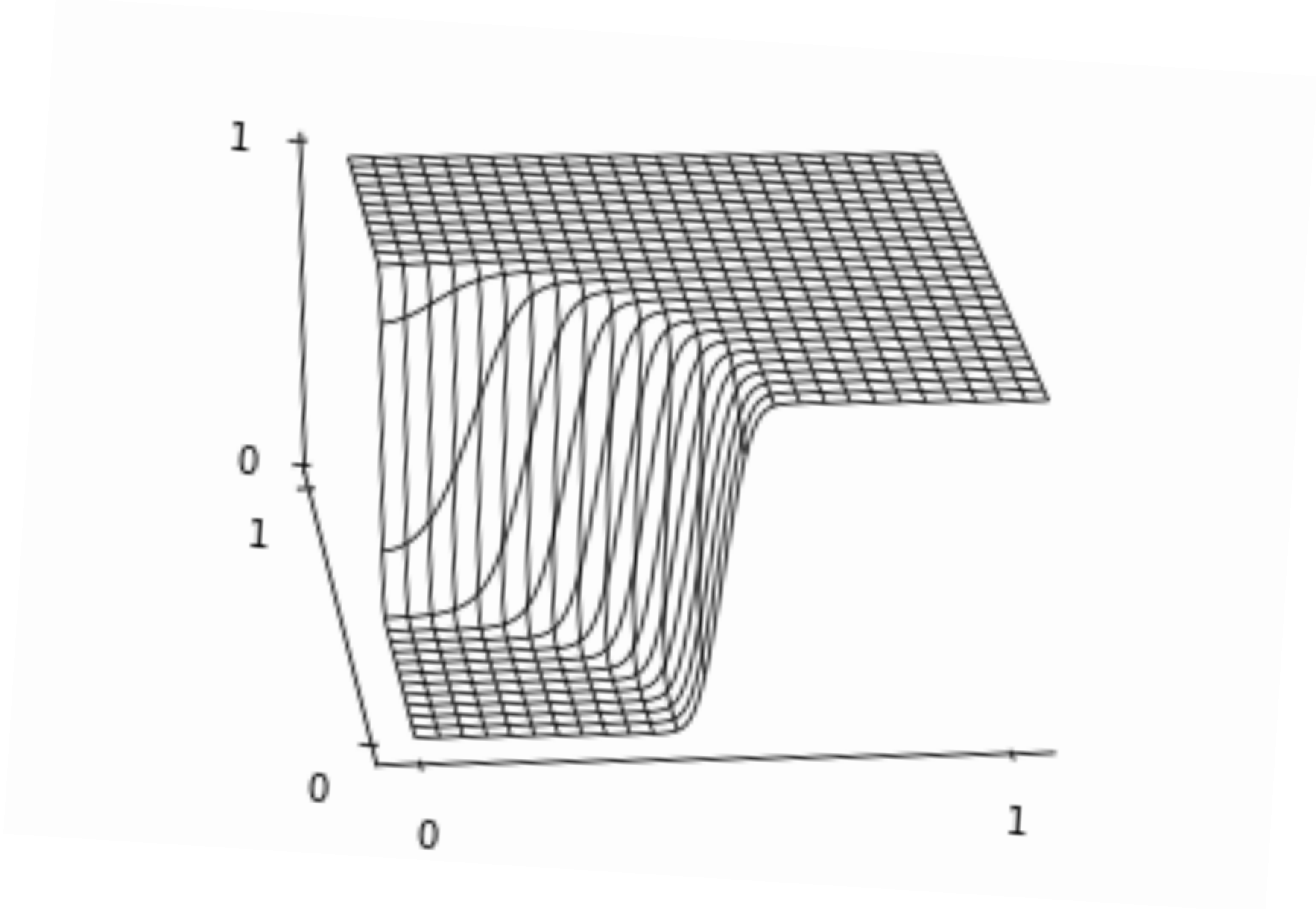
steady-state advection equation:

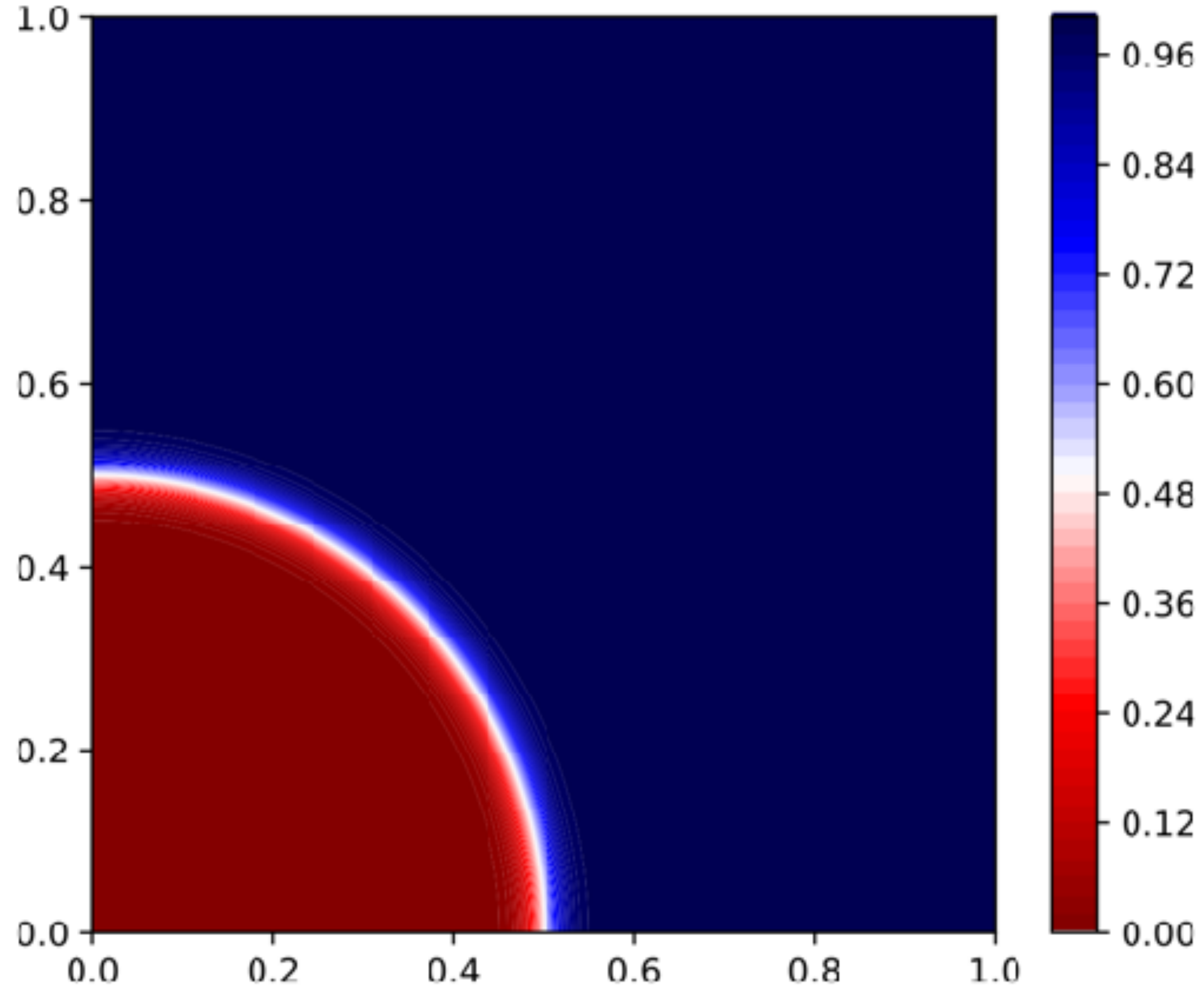
$$\nabla \cdot (\mathbf{c}u) = 0 \qquad \text{in } \Omega = (0,1)^2,$$
$$u = g_D \qquad \text{on } \Gamma_{\text{inflow}}$$

circular, counter-clockwise velocity field
cylindrical gen. of the "smooth jump"

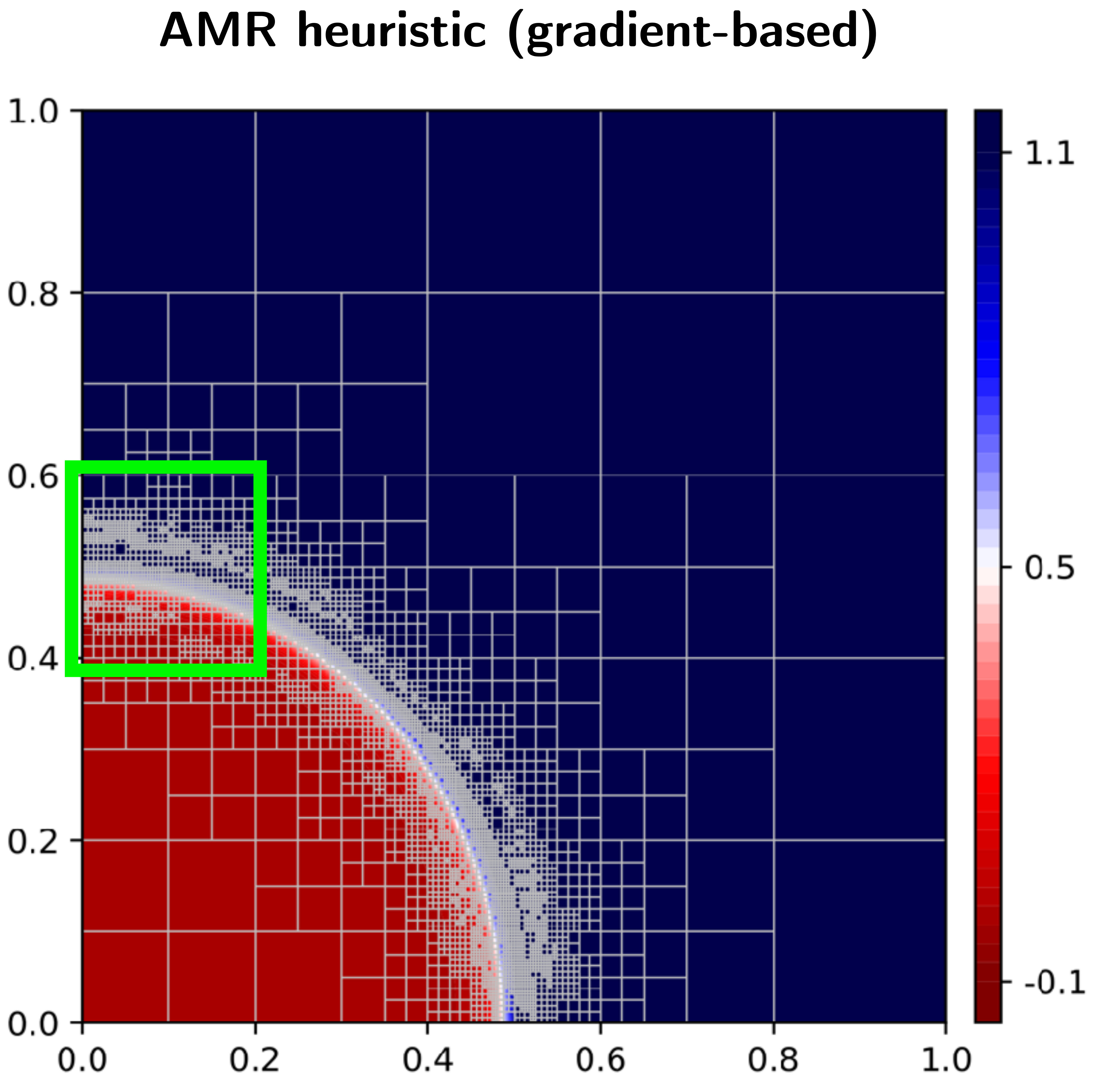
$$\mathbf{c} = \frac{1}{\|x\|_2}(-x_2,\, x_1)$$

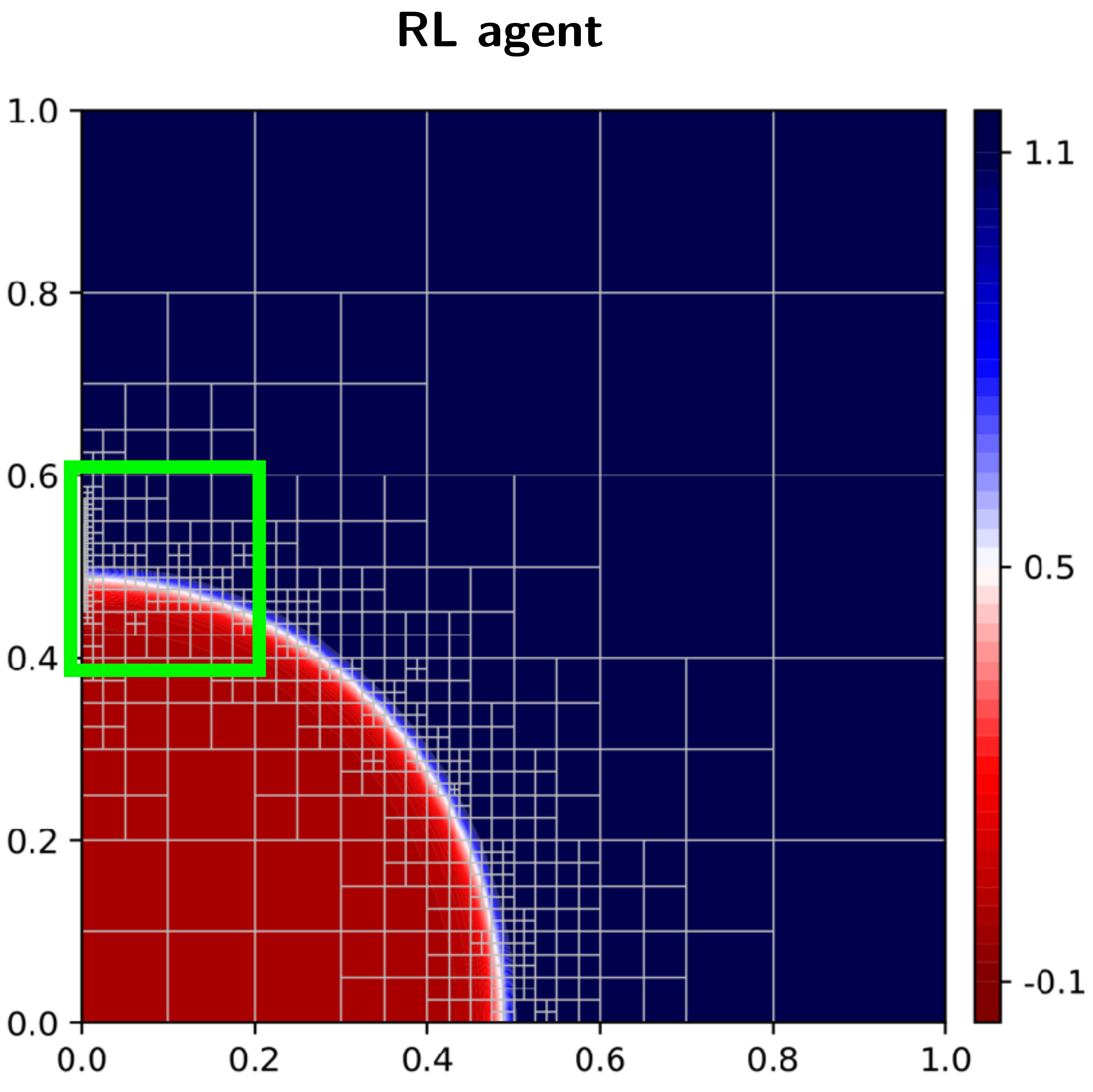
$$g_D = \begin{cases} 1 - \tanh[\alpha(1 - 4(x - x_0))], & x_1 \in [0,1],\, x_2 = 0 \\ 0, & \text{otherwise} \end{cases}$$





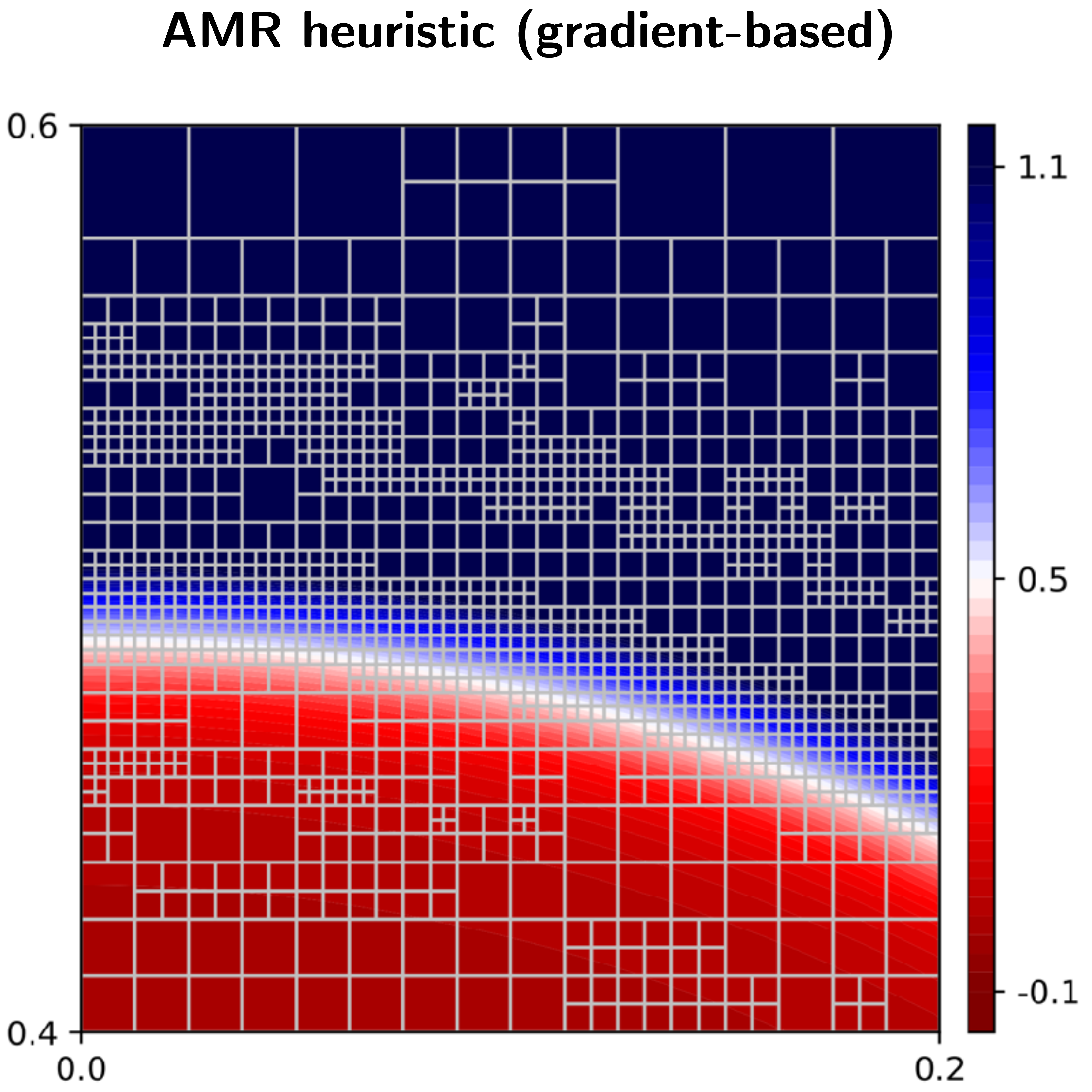
RL agent suggests more conservative refinement

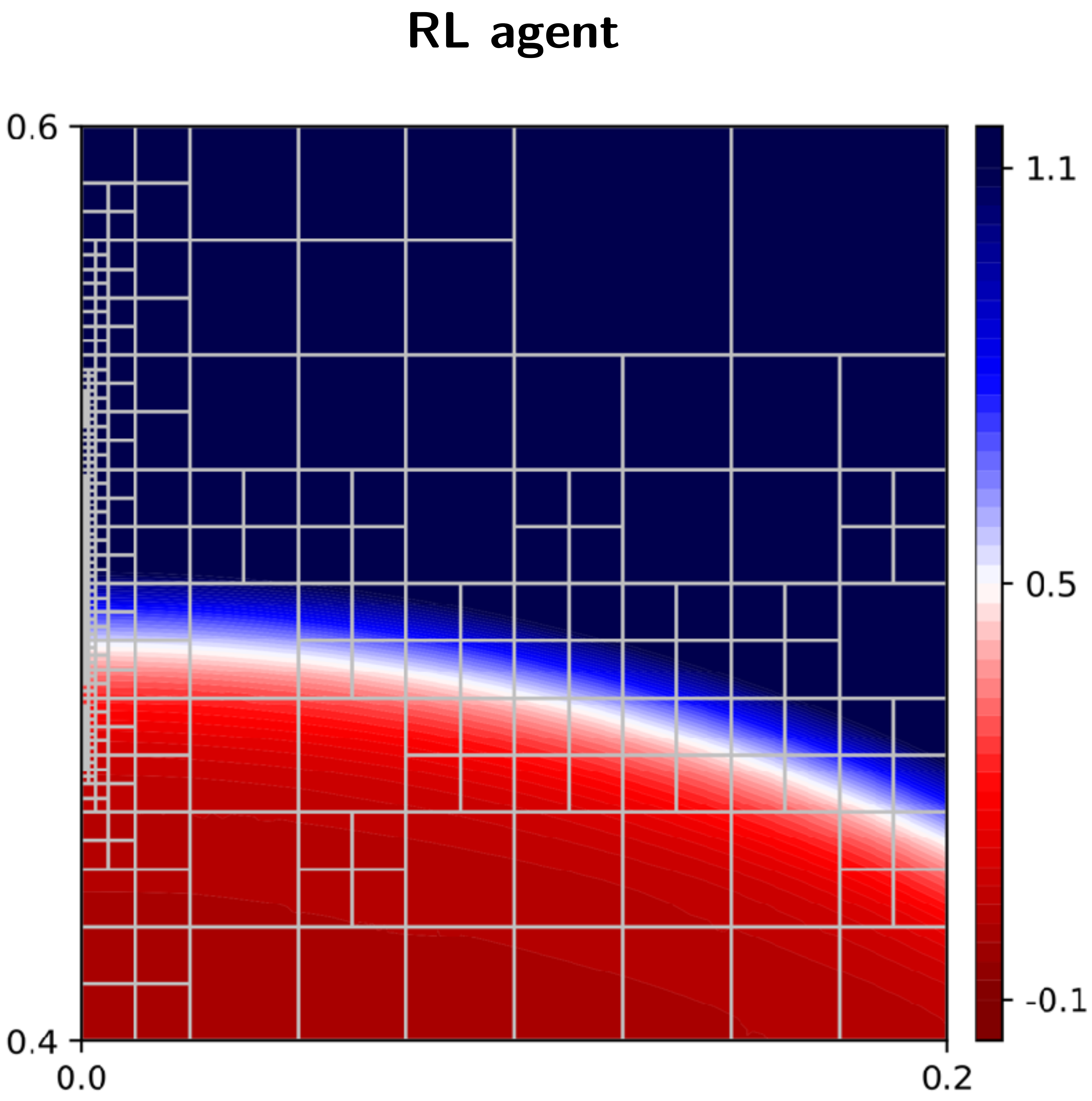




[1] Foucart et al., JCP. 2022.

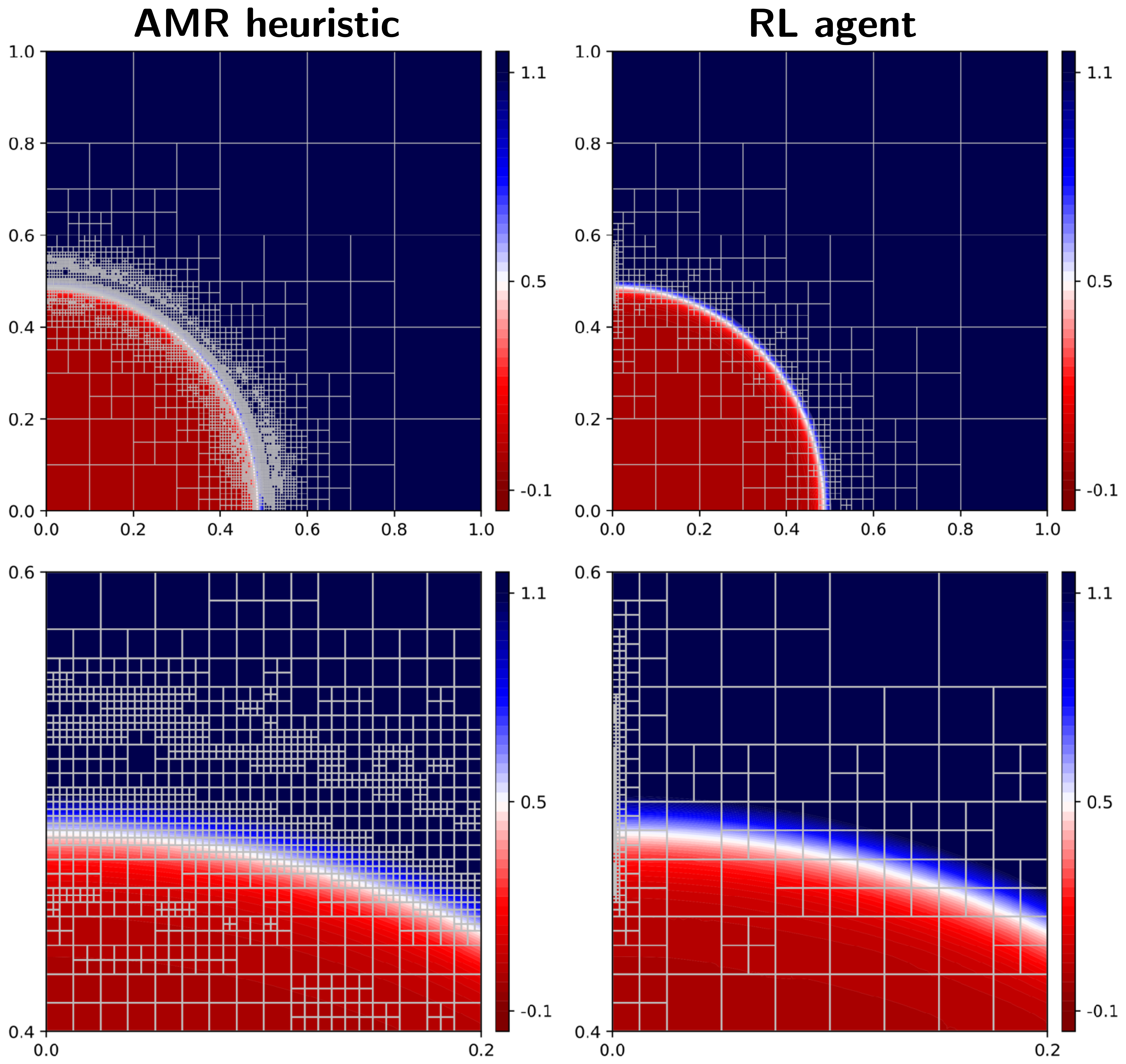
The RL agent's strategy





[1] Foucart et al., JCP. 2022.

How does the RL policy compare to the AMR heuristic?



RL agent learned to refine where the steep gradient meets the outflow boundary

We never "told" it how to do this

In fact, recall that the agent never knows where it "is"

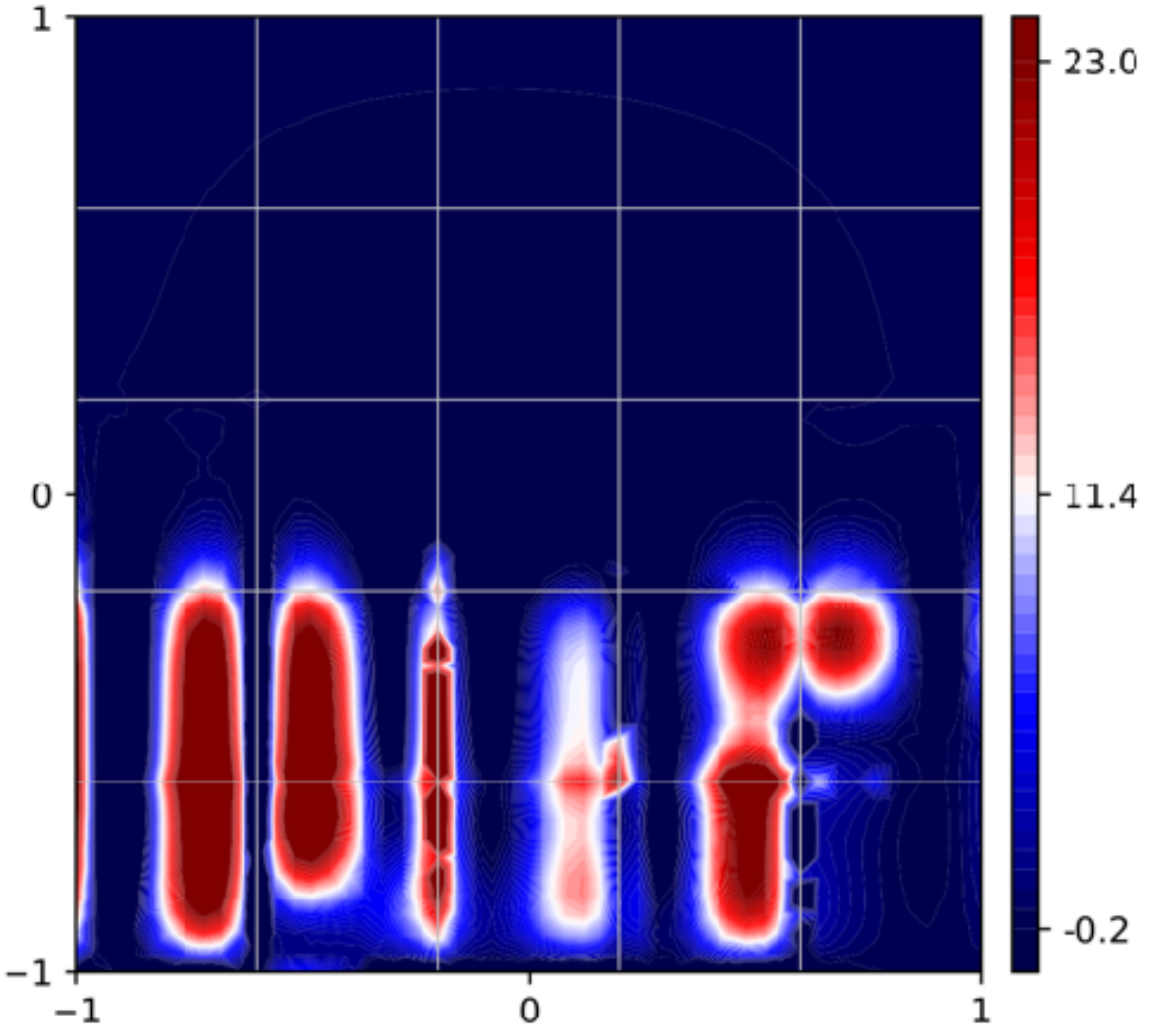
Advances: more complicated PDE, numerical scheme, solution features

Advection diffusion, mixed boundary conditions

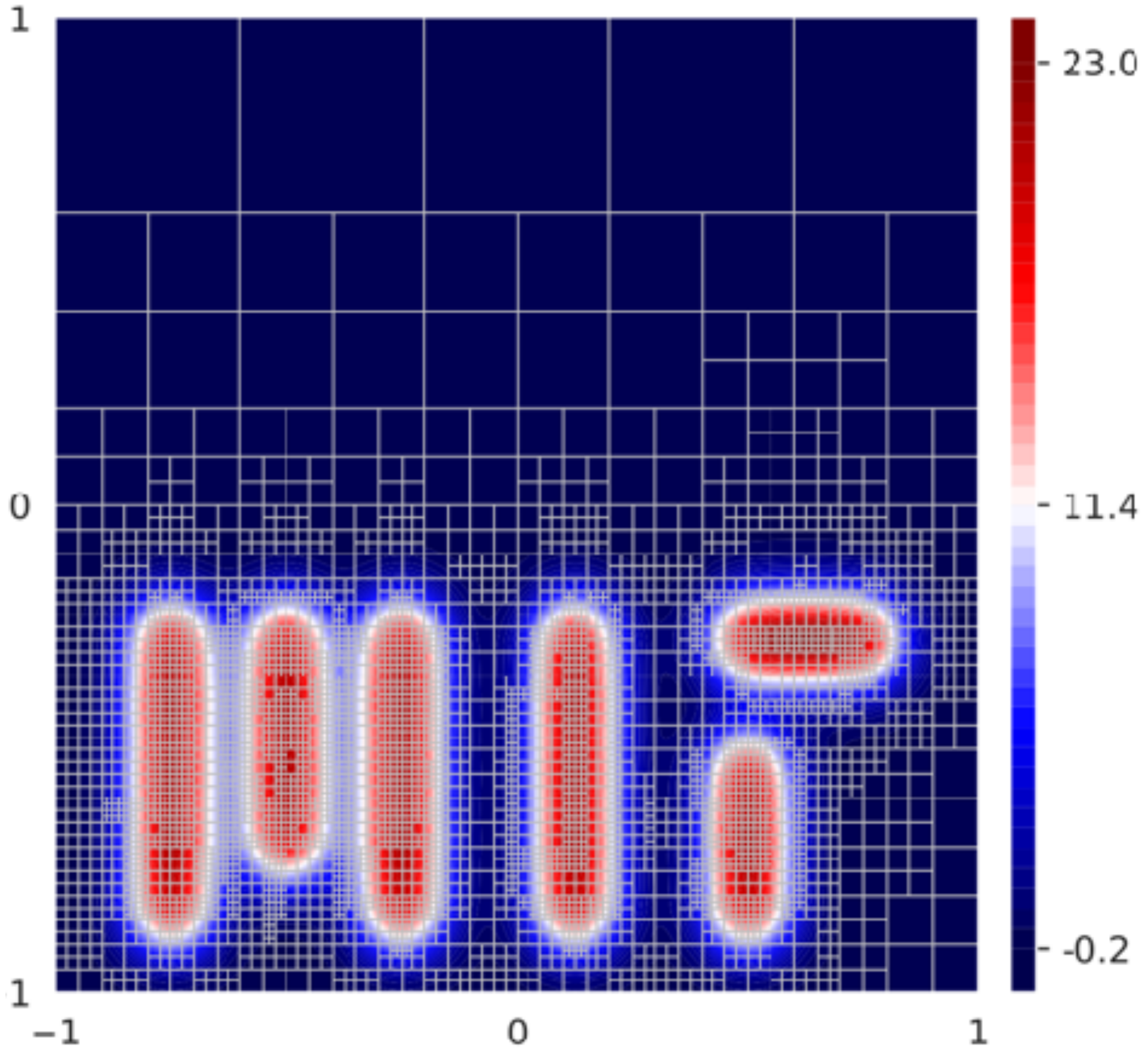
$$
\begin{aligned}
\nabla \cdot (\mathbf{c}u) - \nabla \cdot (\kappa \nabla u) &= f, && \text{in } \Omega \\
(-\kappa \nabla u + \mathbf{c}u) \cdot \mathbf{n} &= g_N, && \text{on } \Gamma_N, \\
u &= g_D, && \text{on } \Gamma_D,
\end{aligned}
$$

Discretized with a fourth-order hybridizable discontinuous Galerkin (HDG) method **[1]**

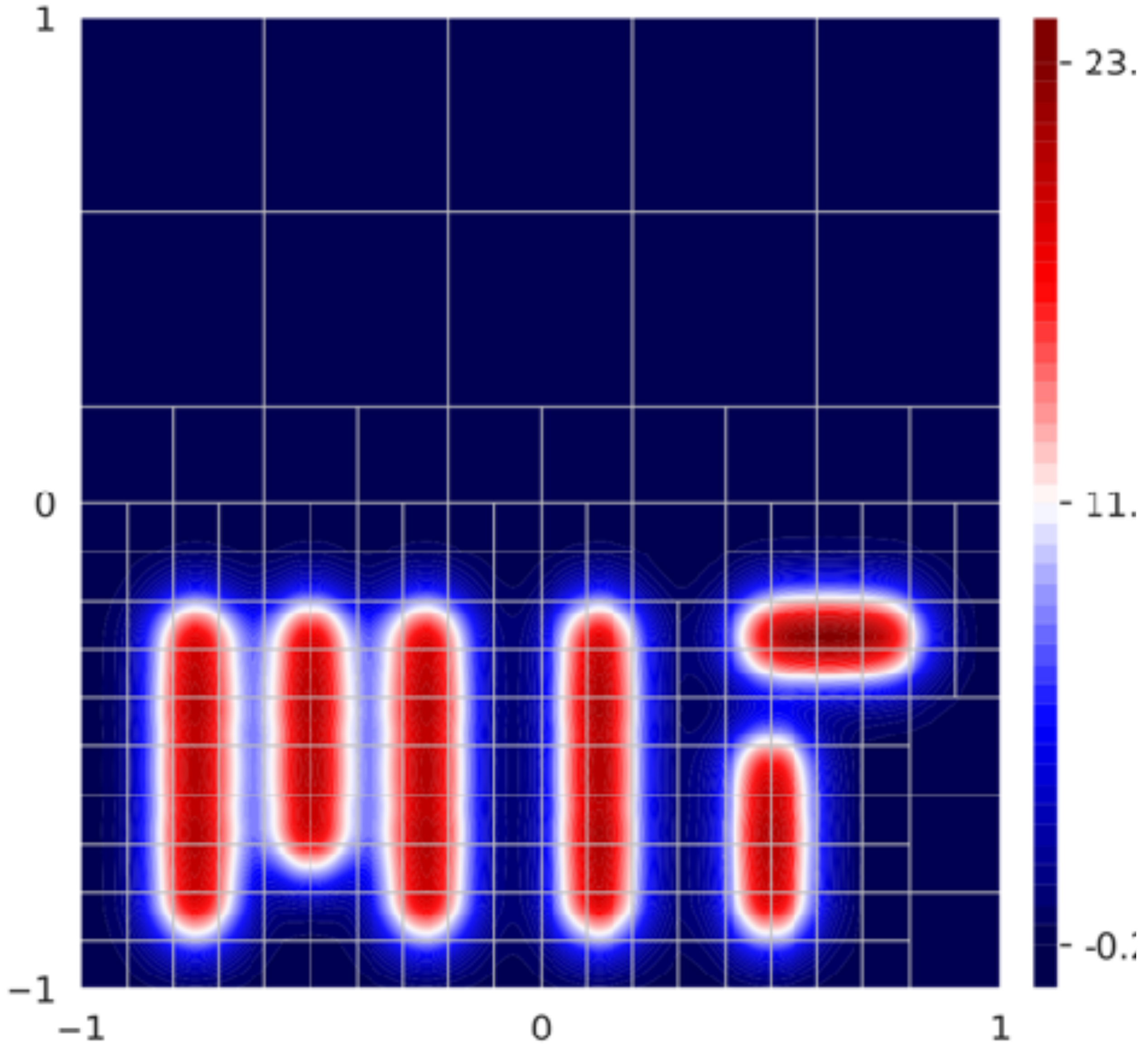
Starting mesh



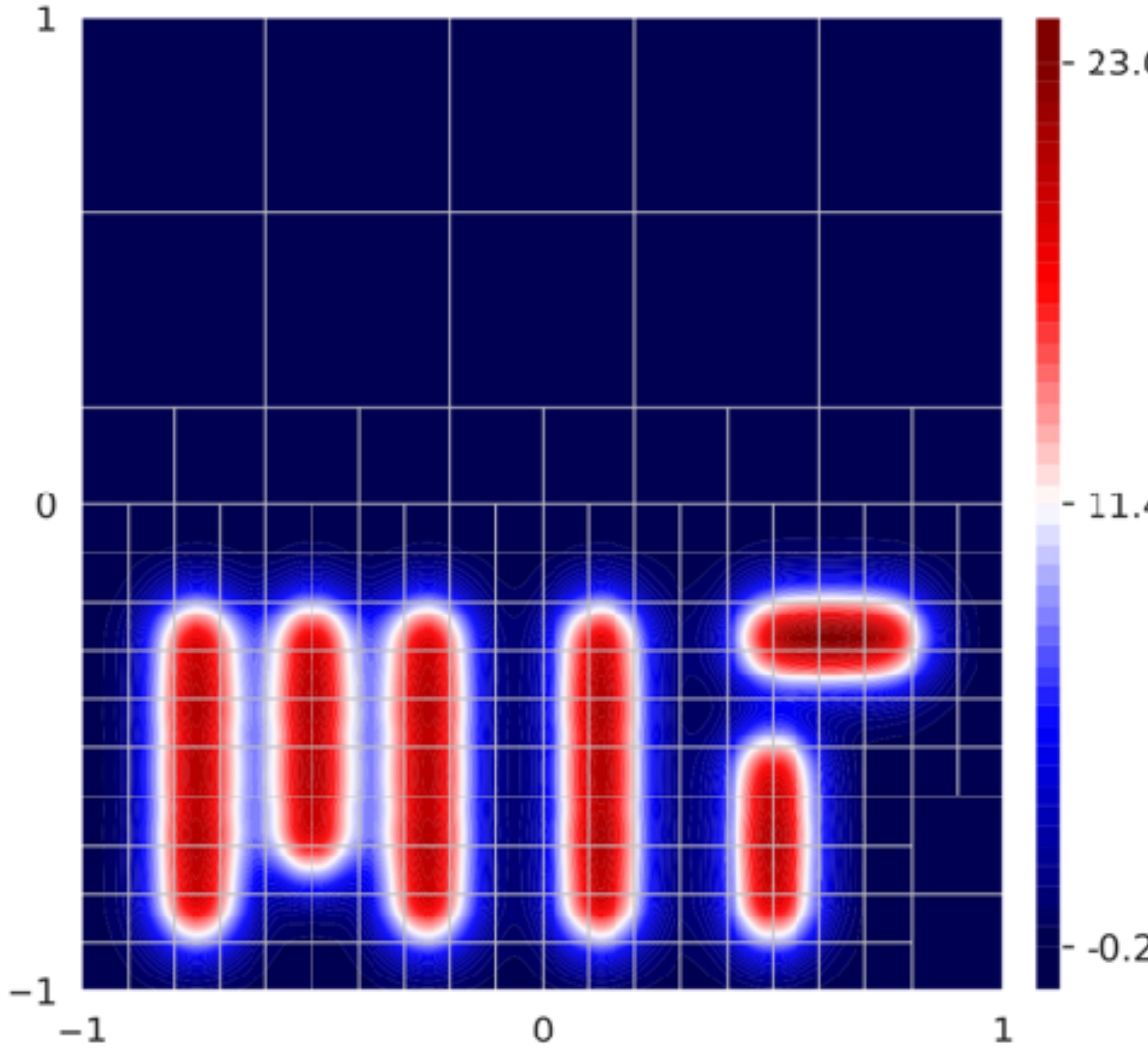
AMR Heuristic



RL agent (1500 cell)



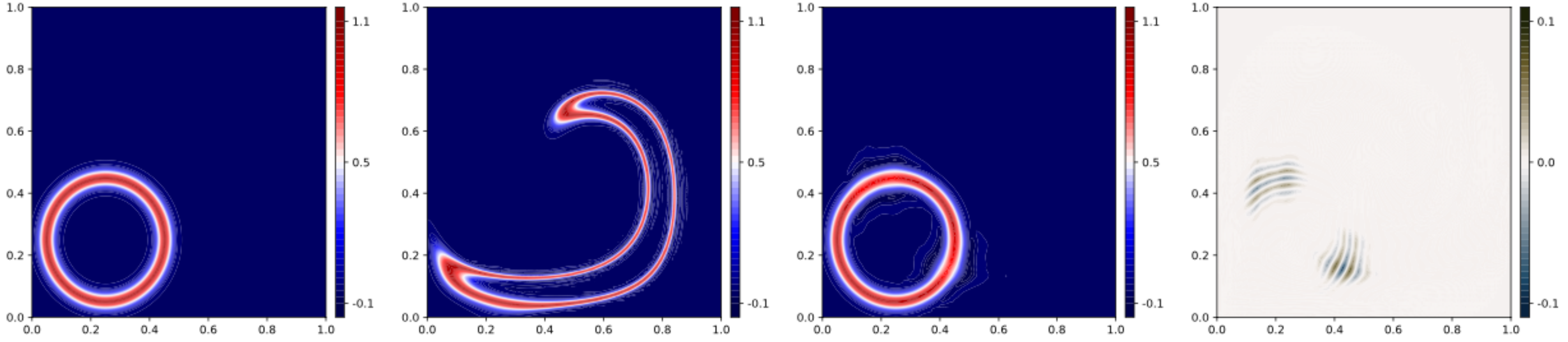
RL agent (5000 cell)



[1] Nguyen et al., JCP 2009a. [2] Foucart et al., JCP. 2022.

More complicated, time-dependent dynamics

2D ring advection [1] test case:

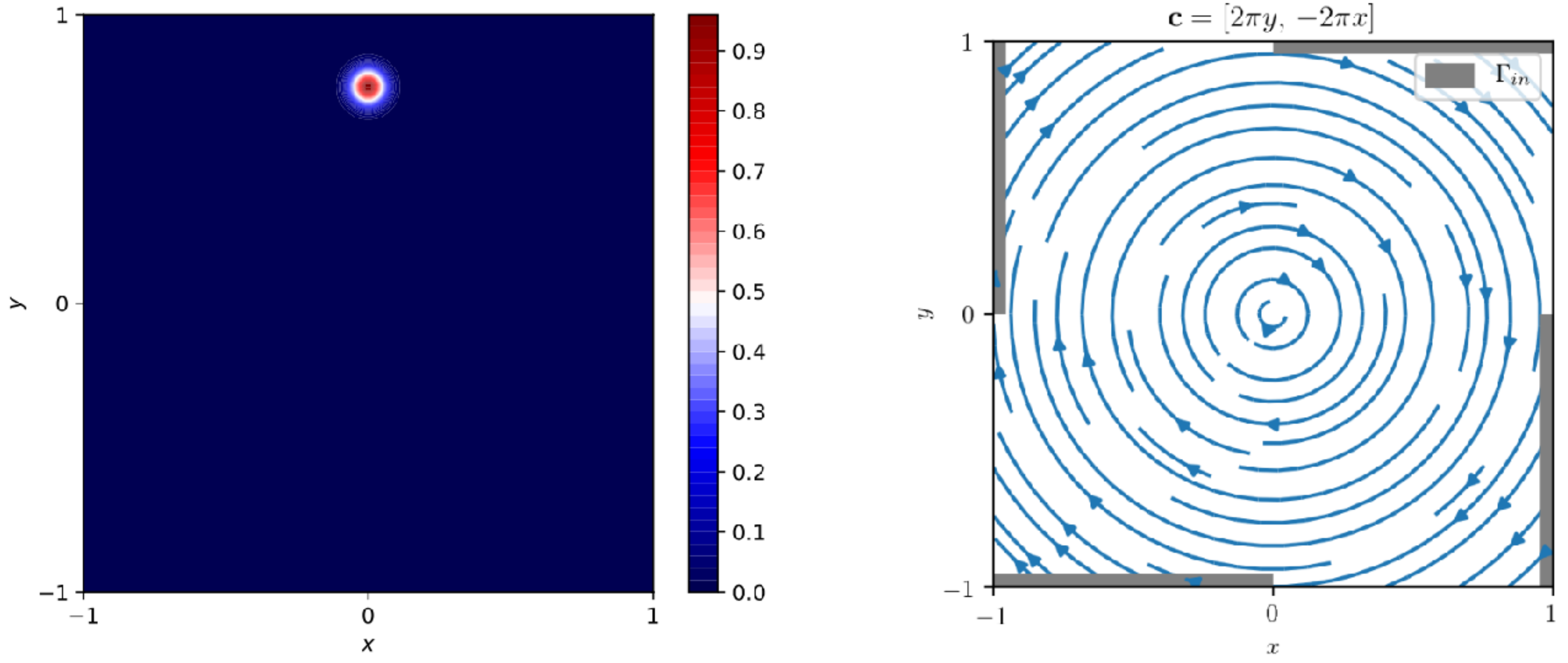


(a) $t = 0$ (b) $t = 0.5$ (c) $t = 1$ (d) $u_h - u_{\text{exact}}$, $t = 1$

Figure 19: Unsteady reversible 2D ring advection. Numerical solution on a 64×64 uniform grid.

More complicated time-dependent dynamics

Train on a much simpler example: advection of Gaussian pulse

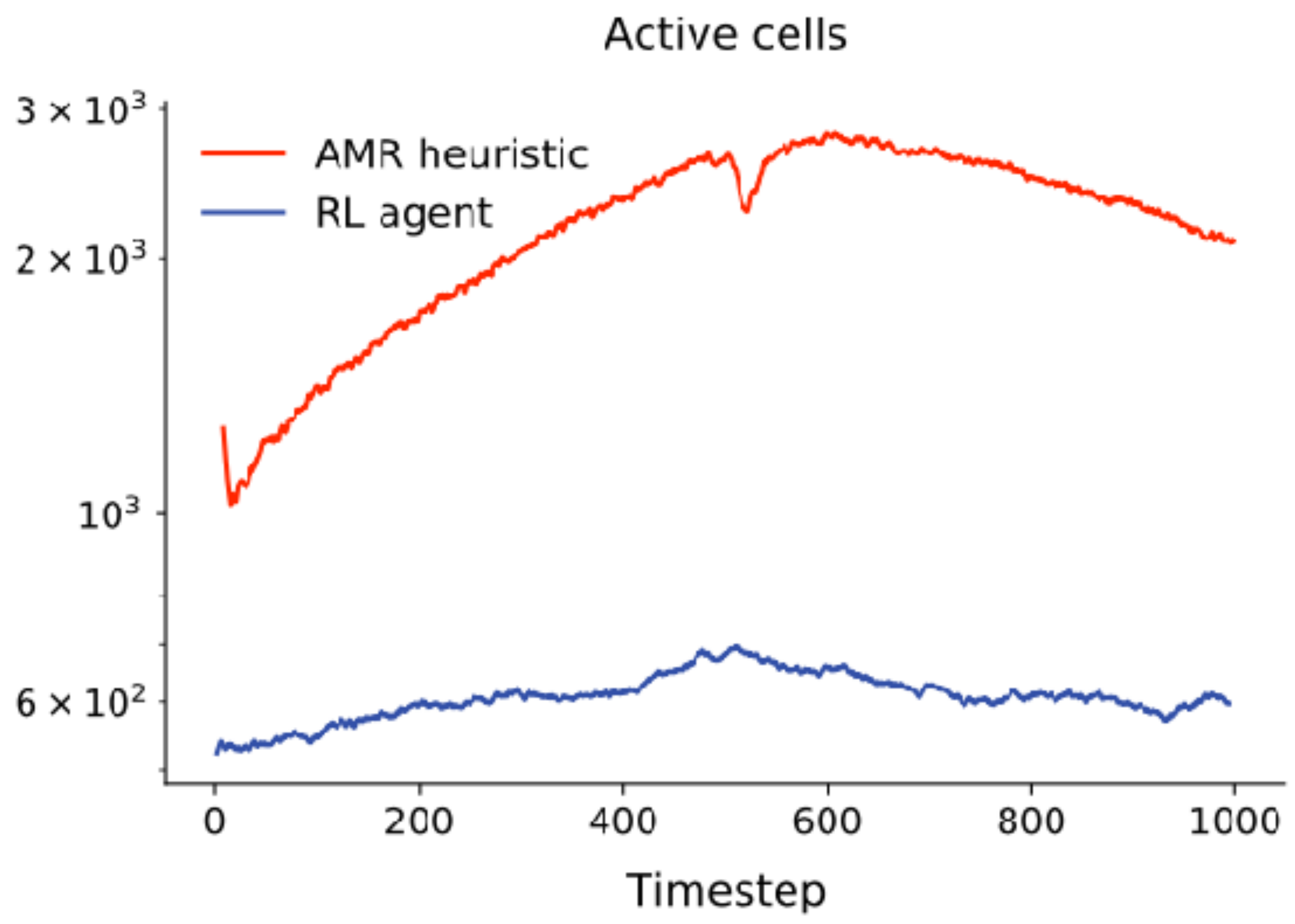


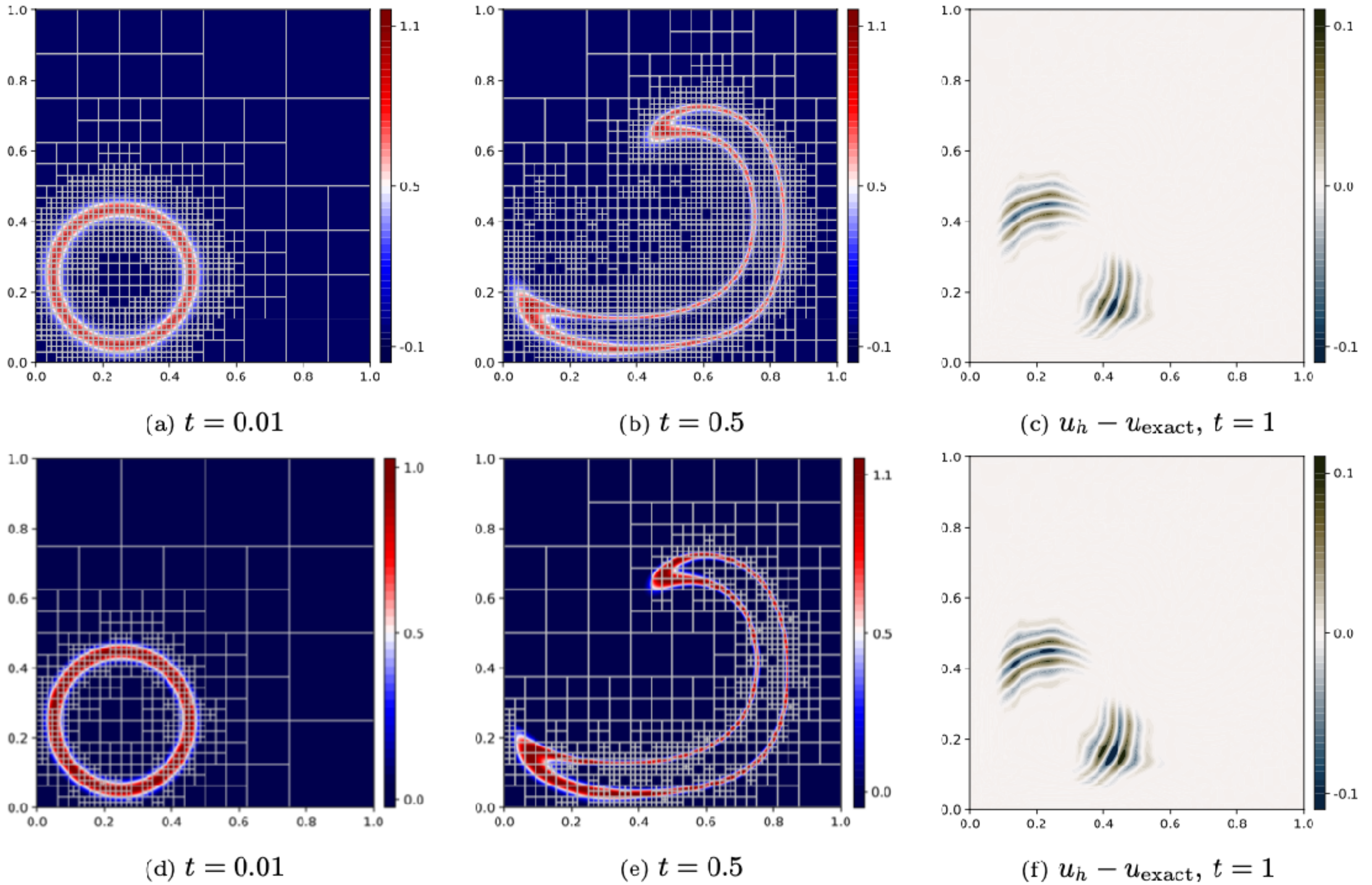
(a) Initial condition u_0

(b) Convective velocity $\boldsymbol{c}$

More complicated time-dependent dynamics: 2D ring advection

- The RL agent is able to accurately preserve the shape of the ring over time integration
- Practically same accuracy as the AMR heuristic (Kelly bulk refinement), but does so at a fraction of the cost of the heuristic in terms of number of cells





(a) $t = 0.01$ (b) $t = 0.5$ (c) $u_h - u_{\text{exact}},\ t = 1$

(d) $t = 0.01$ (e) $t = 0.5$ (f) $u_h - u_{\text{exact}},\ t = 1$

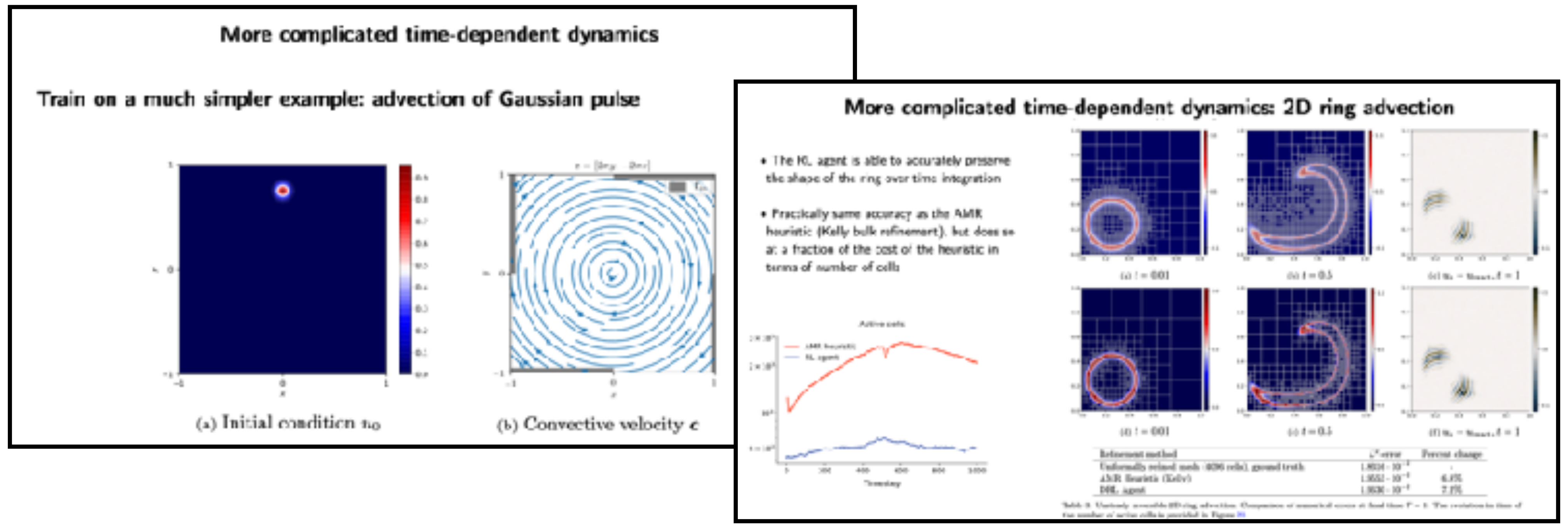
Refinement method	L^2-error	Percent change
Uniformally refined mesh (4096 cells), ground truth	$1.8316 \cdot 10^{-2}$	-
AMR Heuristic (Kelly)	$1.9553 \cdot 10^{-2}$	6.8%
DRL Agent	$1.9630 \cdot 10^{-2}$	7.2%

Table 3: Unsteady reversible 2D ring advection. Comparison of numerical errors at final time $T = 1$. The evolution in time of the number of active cells is provided in Figure 21.

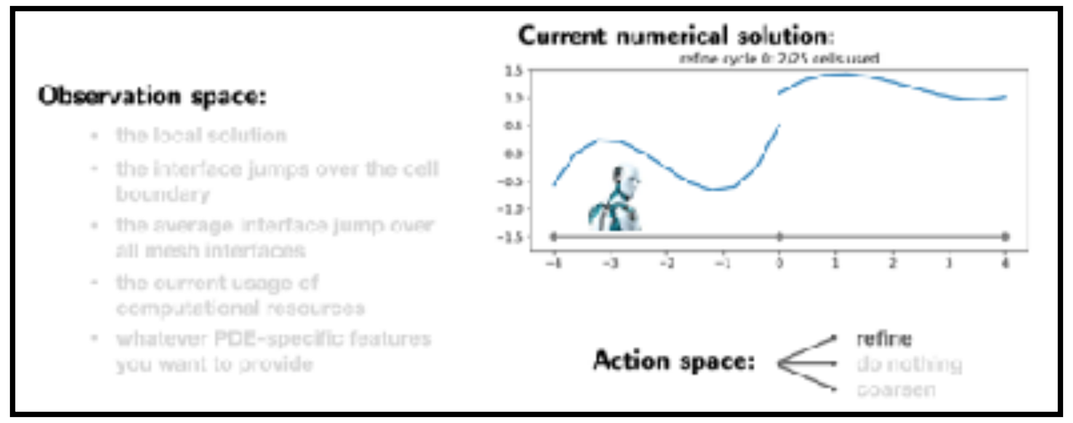
Deep RL for AMR: summary

Effective AMR policies can be discovered directly from numerical simulation using deep reinforcement learning.

- the RL policy network is trained directly through experience by numerical simulation on small problems, and can be deployed on much larger problems

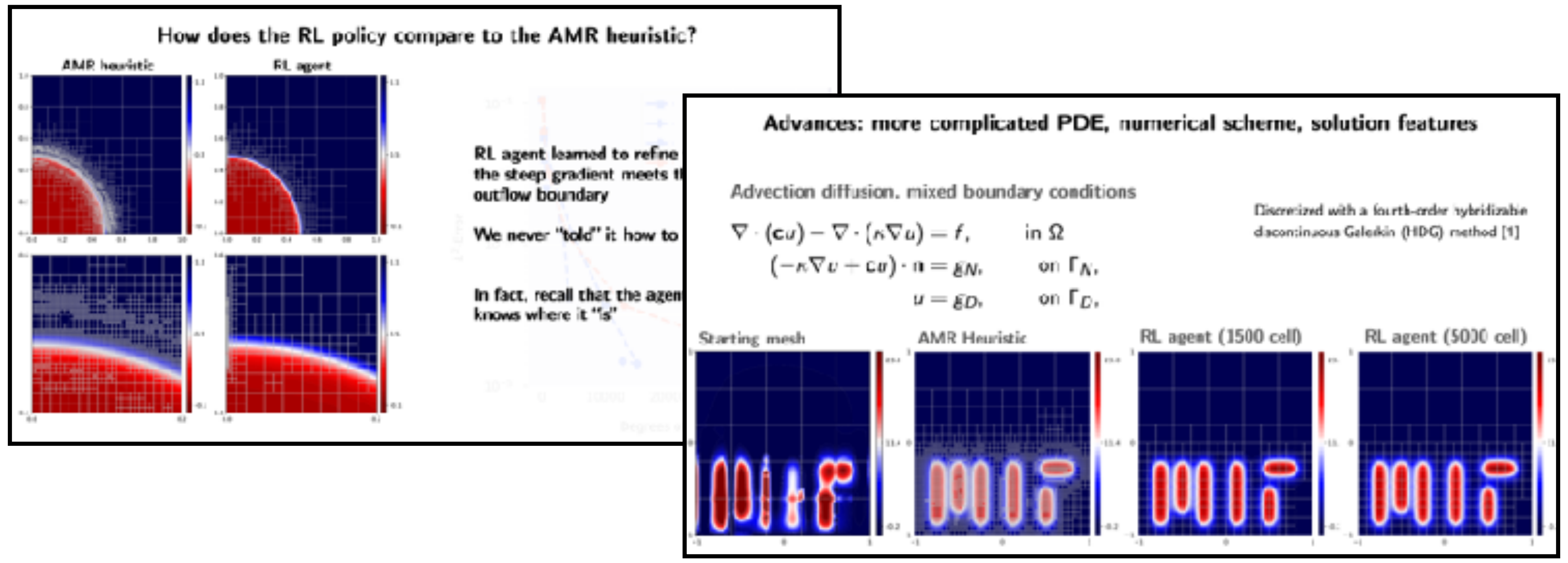


- learns a local relationship, and therefore robust to overfitting on any particular feature seen during training



- the deep-RL AMR technique is not specific to any dimension, PDE, or numerical solver

- at no point during training or model deployment do we ever use an exact solution or "ground truth"



Publications

Scientific machine learning:

Foucart, C., A. Charous, and P.F.J. Lermusiaux, 2022. Deep Reinforcement Learning for Adaptive Mesh Refinement. *Journal of Computational Physics*,. arXiv preprint

Charous, A., **C. Foucart**, and P.F.J. Lermusiaux, 2023. Automatically differentiable and learnable coordinate transforms. **[in prep]**

DG-FEM, ocean modeling:

Foucart, C., C. Mirabito, A. Karthik, P.J. Haley, Jr., and P.F.J. Lermusiaux, 2023a. A Hybridizable Discontinuous Galerkin Nonhydrostatic Ocean Model with a Free Surface. *Ocean Modelling*, **[in prep]**

Foucart, C., C. Mirabito, P.F.J. Lermusiaux, 2023b. Stable and Robust Projection Methods for the Incompressible Navier-Stokes Equations using Hybridizable Discontinuous Galerkin Schemes, **[in prep]**

Foucart, C., C. Mirabito, P.J. Haley, Jr., and P.F.J. Lermusiaux, 2021. High-order Discontinuous Galerkin Methods for Nonhydrostatic Ocean Processes with a Free Surface. In: OCEANS '21 IEEE/MTS San Diego, 20-23 September 2021, pp. 1-9.

Foucart, C., C. Mirabito, P. J. Haley, Jr., and P. F. J. Lermusiaux, 2018. Distributed Implementation and Verification of Hybridizable Discontinuous Galerkin Methods for Nonhydrostatic Ocean Processes. *Oceans ’18 MTS/IEEE* Charleston, 22-25 October 2018.

Foucart, C., 2019. Efficient Matrix-Free Implementation and Automated Verification of Hybridizable Discontinuous Galerkin Finite Element Methods. SM Thesis, Massachusetts Institute of Technology, Mechanical Engineering, June 2019.

Other contributions:

Ali, W.H., Y. Gao, **C. Foucart**, M. Doshi, C. Mirabito, P.J. Haley, and P.F.J. Lermusiaux, 2022. High-Performance Visualization for Ocean Modeling. *OCEANS '22 IEEE/MTS* Hampton Roads, 17-21 October 2022.

Ali, W.H., M.H. Mirhi, A. Gupta, C.S., **C. Foucart**, M.M. Doshi, D.N. Subramani, C. Mirabito, P.J. Haley, Jr., and P.F.J. Lermusiaux, 2019. SeaVizKit: Interactive Maps for Ocean Visualization. *OCEANS '19 MTS/IEEE* Seattle, 27-31 October 2019.

Deluca, S., B. Rocchio, **C. Foucart**, C. Mirabito, S. Zanforlin, P.J. Haley, and P.F.J. Lermusiaux, 2018. Scalable Coupled Ocean and Water Turbine Modeling for Assessing Ocean Energy Extraction. *Oceans ’18 MTS/IEEE* Charleston, 22-25 October 2018.

Summary of contributions

Nonhydrostatic HDG Ocean Modeling [2], [3]

- Development and implementation of a novel HDG spatial discretization of the hydrostatic & nonhydrostatic ocean equations with a free surface
- Wrote a parallelized C++ library implementation
- Verification of NHS / HS model with results from linear wave theory
- Idealized simulations of nonhydrostatic behavior (subduction, internal solitary waves)
- Implemented model nesting in the MSEAS-PE code for use with real ocean data in 2D and 3D
- Conducted preliminary investigations of nonhydrostatic behavior in the Alboran Sea

Deep Reinforcement Learning for AMR [1]

- Formulated adaptive mesh refinement as a reinforcement learning problem
- Conceived and implemented neural network architectures to learn to solve the reinforcement learning problem, without training data or an exact solution
- Wrote a binding language between custom AMR-capable DG/HDG solvers and machine learning libraries
- Showed the resultant trained policies to be competitive or better than many common AMR heuristics, over a wide range of PDEs and test cases

Miscellaneous DG/HDG contributions:

- Quadrature-free HDG operator formulation & matrix-free schemes [5]
- Computational considerations for treatment of the singular Poisson equation
- UQ: EnKF for HDG ensembles, Bayesian inversion of PDE coefficients using MCMC with DG schemes
- Parallel and multi-threaded computing approaches for HDG schemes [4], [5]

[1] Foucart et al., JCP, 2022. [2] Foucart et al., Ocean Modelling. in prep. 2023a. [3] Foucart et al., IEEE Oceans 2021.
[4] Foucart et al. IEEE Oceans, 2018. [5] Foucart, SM Thesis, 2019.

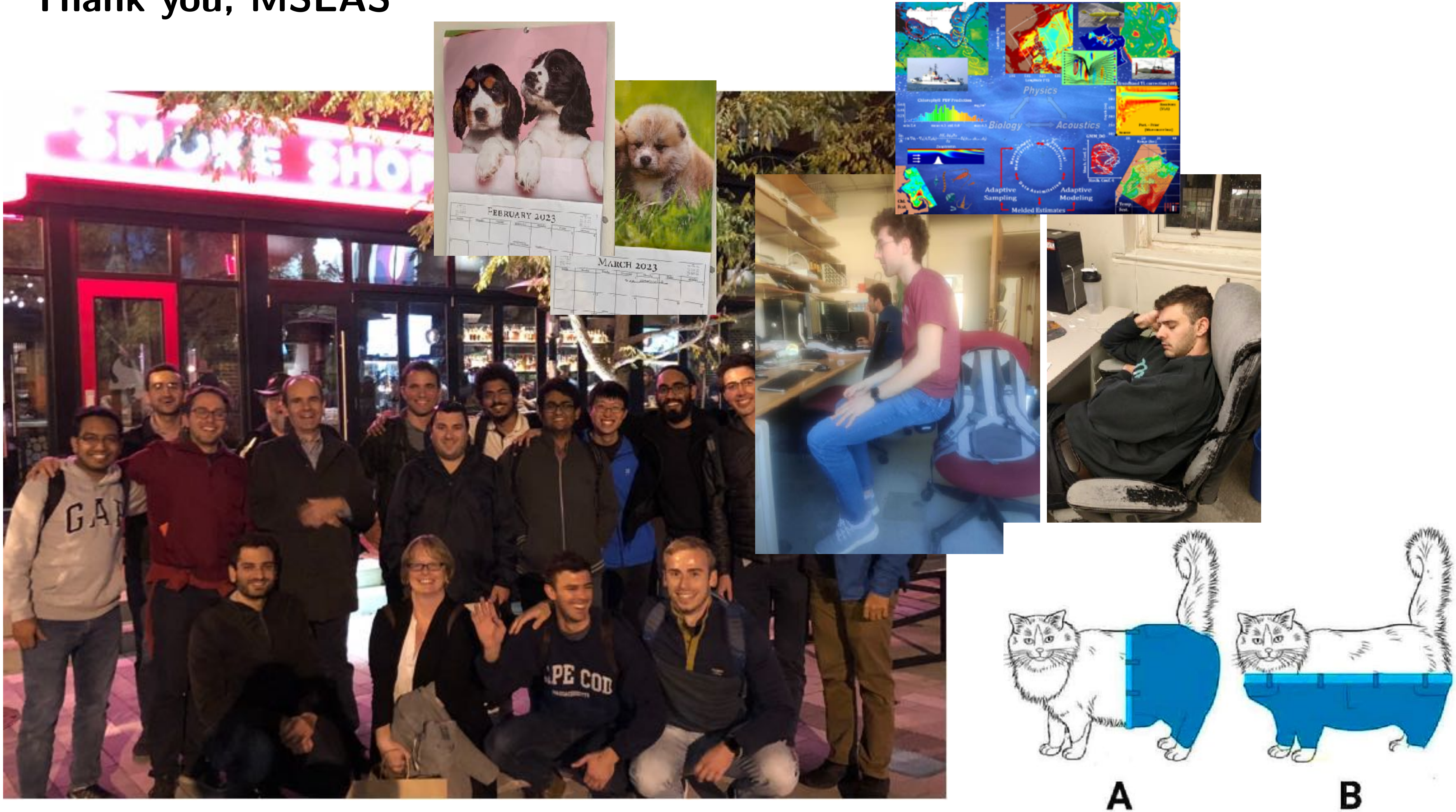
Acknowledgements

- **Committee Members:** Prof. Pierre F.J. Lermusiaux, Dr. Cuong Nguyen, Prof. Anthony Patera, Prof. Nicholas Patrikalakis, Prof. Jaime Peraire

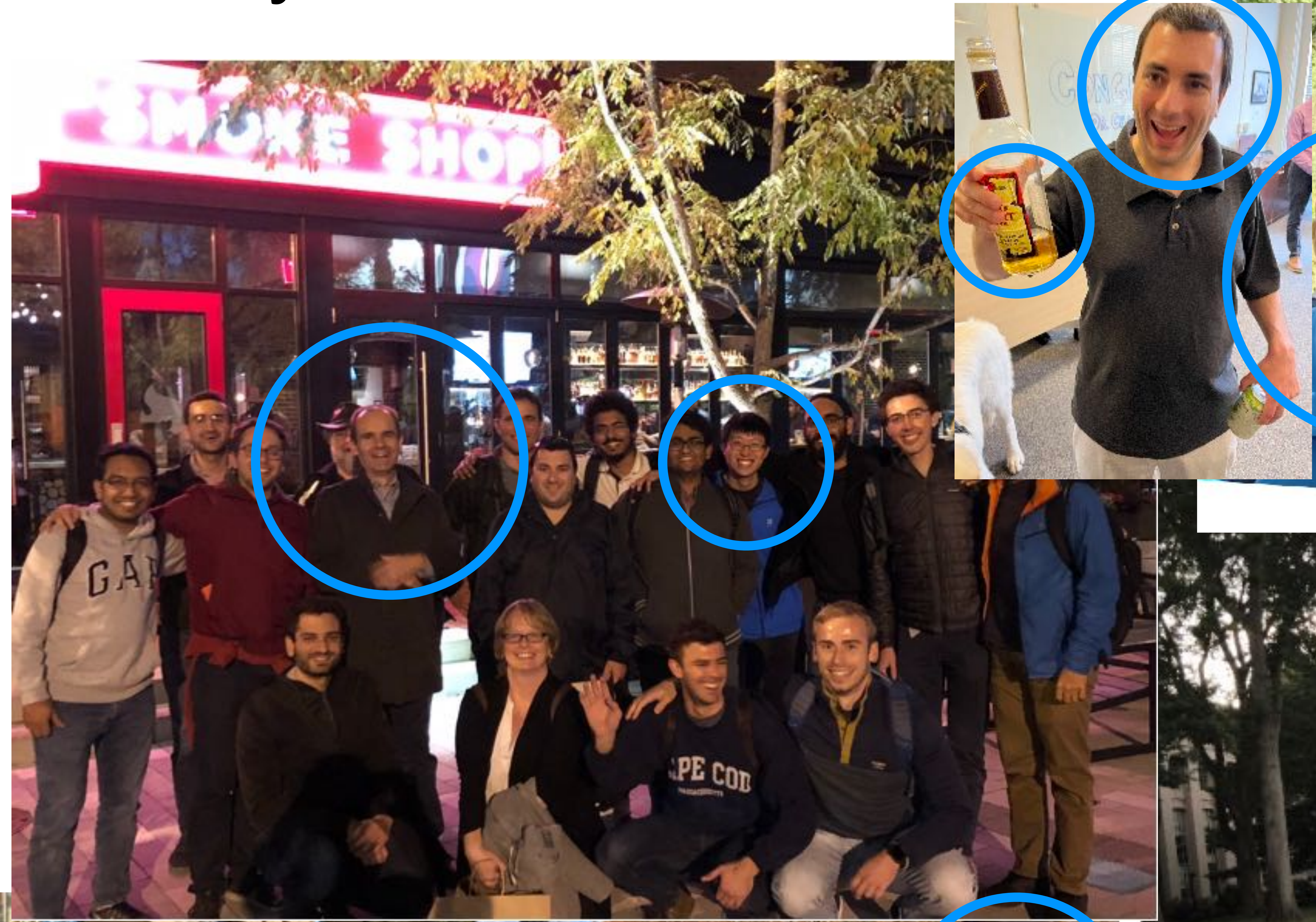
- **Funding:** Office of Naval Research for support throughout this research and Ph.D. under grants N00014-15-1-2626 (DRI-FLEAT), N00014-18-1-2781 (DRI-CALYPSO), and N00014-20-1-2023 (MURI ML- SCOPE), the Defense Advanced Research Projects Agency (DARPA) for support under grant N66001-16-C- 4003 (POSYDON), and the National Science Foundation for support under grants OCE-1061160 (ShelfIT) and EAR-1520825 (NSF-ALPHA)

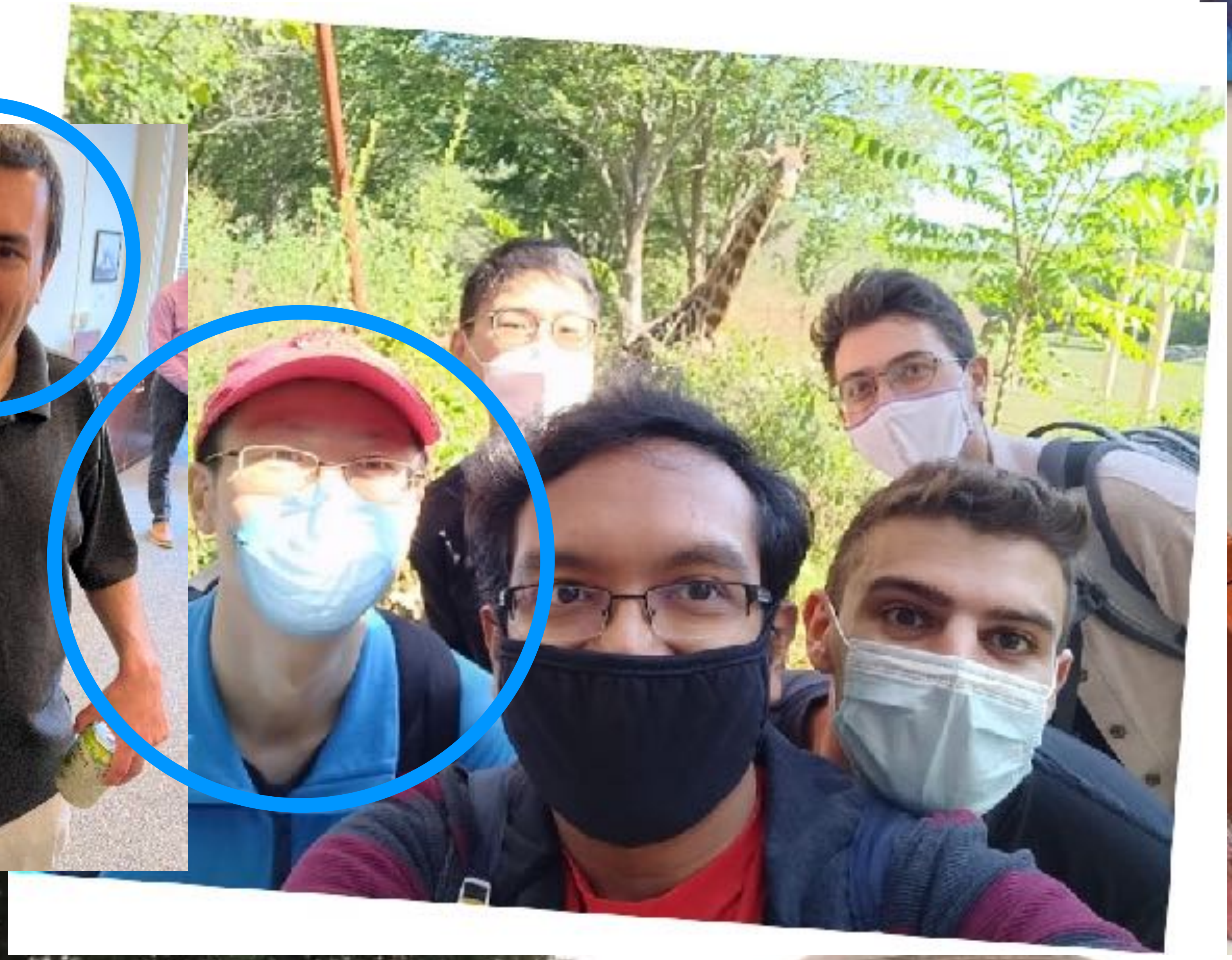
- **Administrative staff:** Lisa Maxwell, Leslie Regan, Una Sheehan, Saana McDaniel, Kate Nelson

Thank you, MSEAS

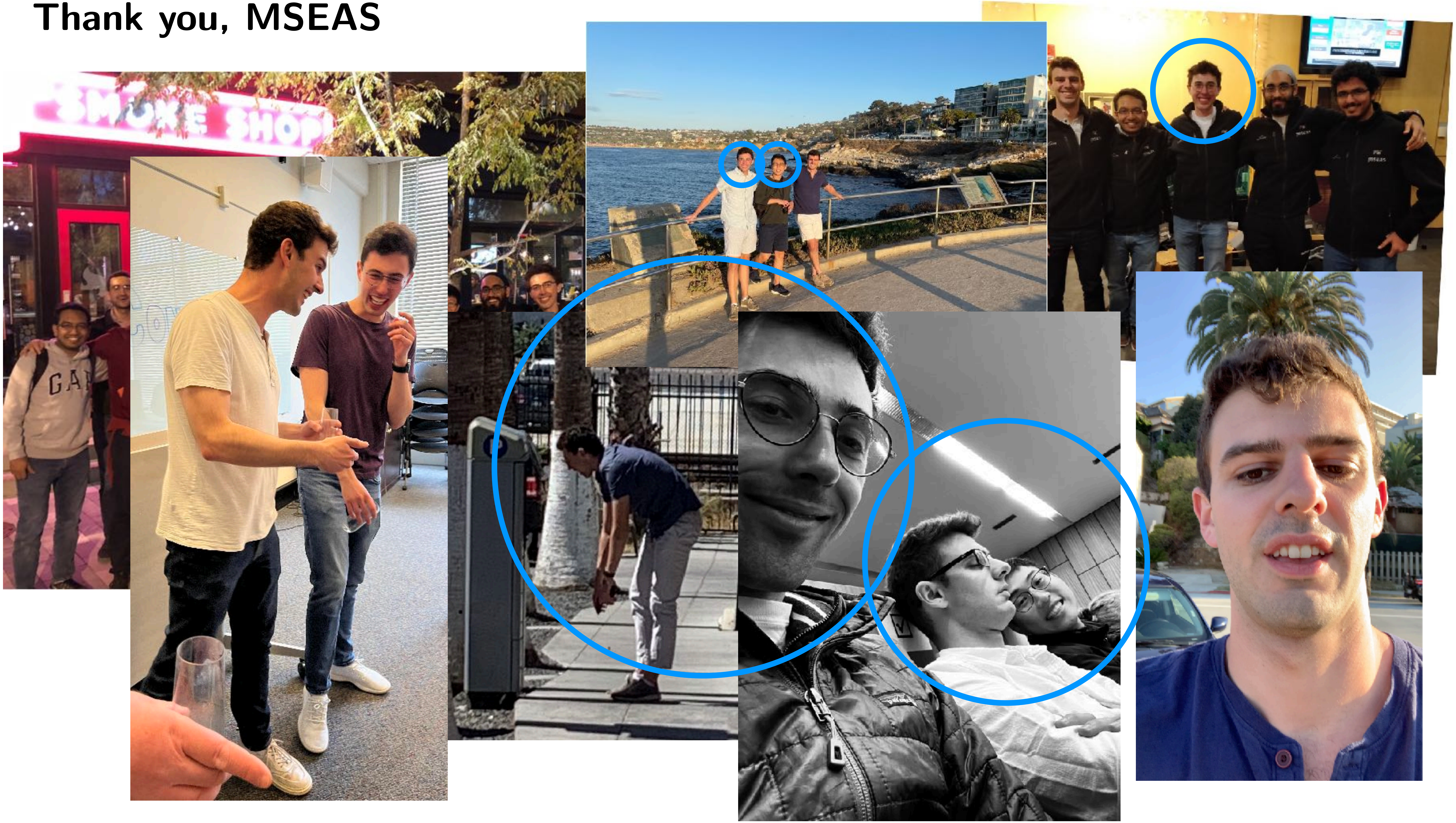


Thank you, MSEAS

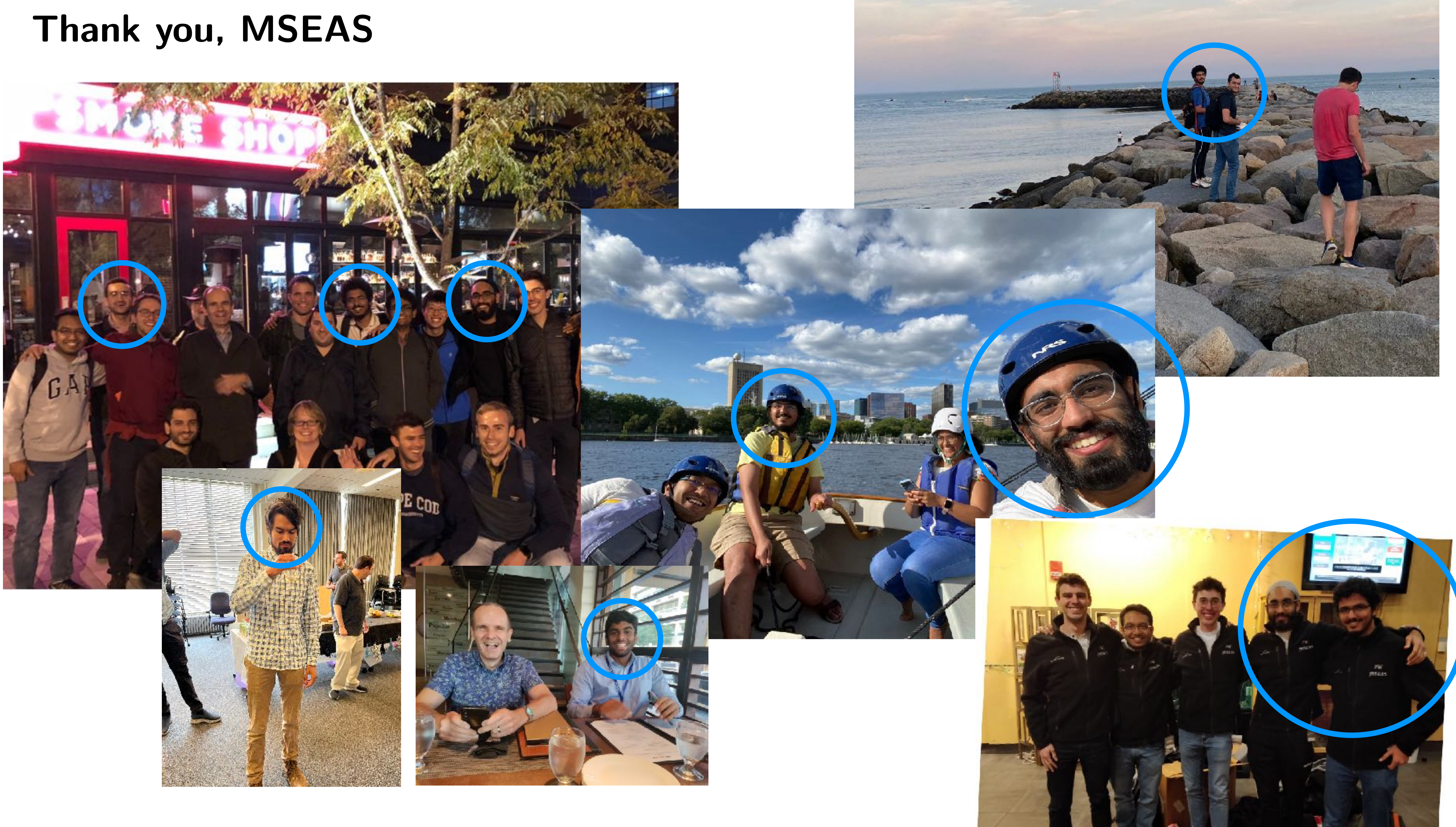




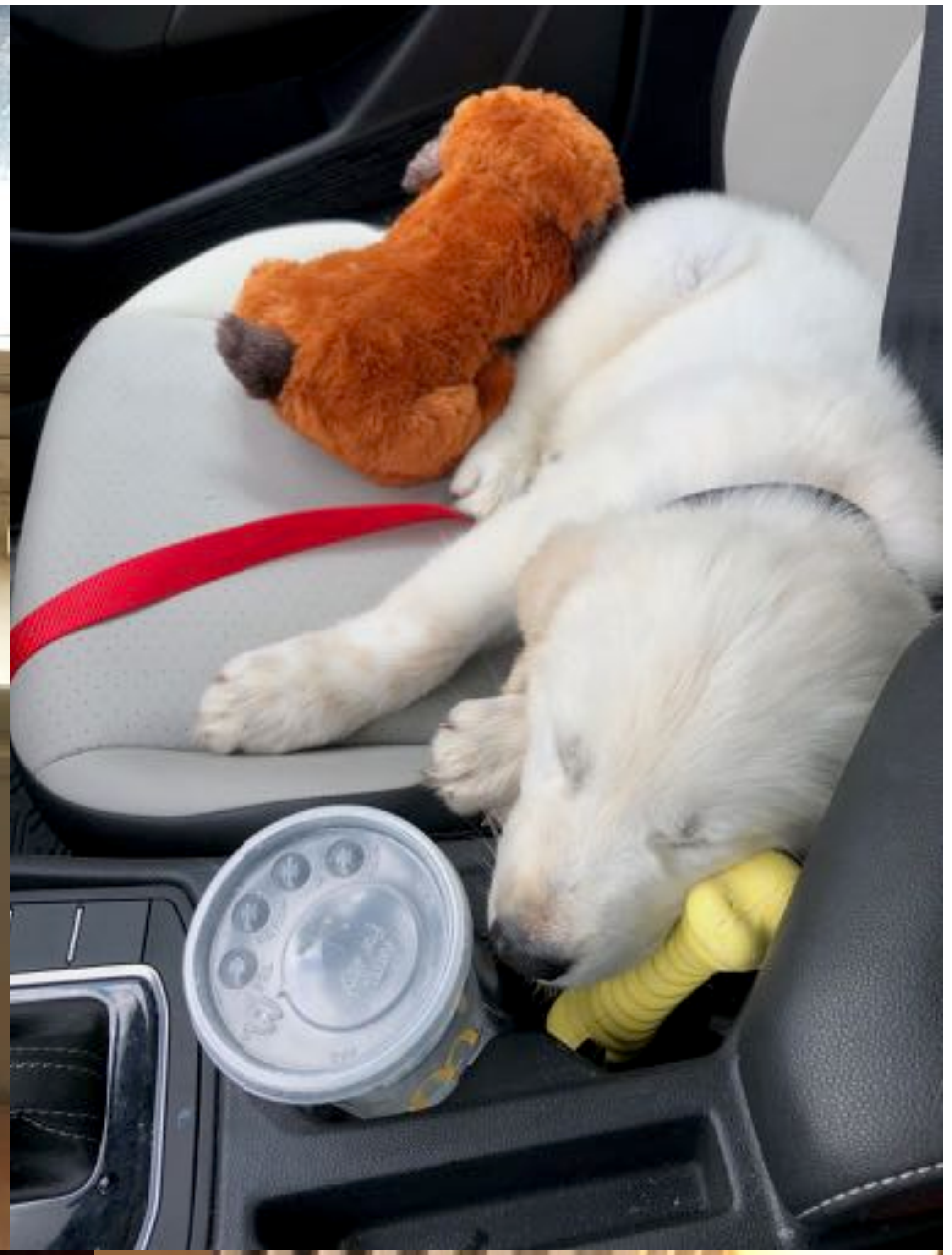
Thank you, MSEAS

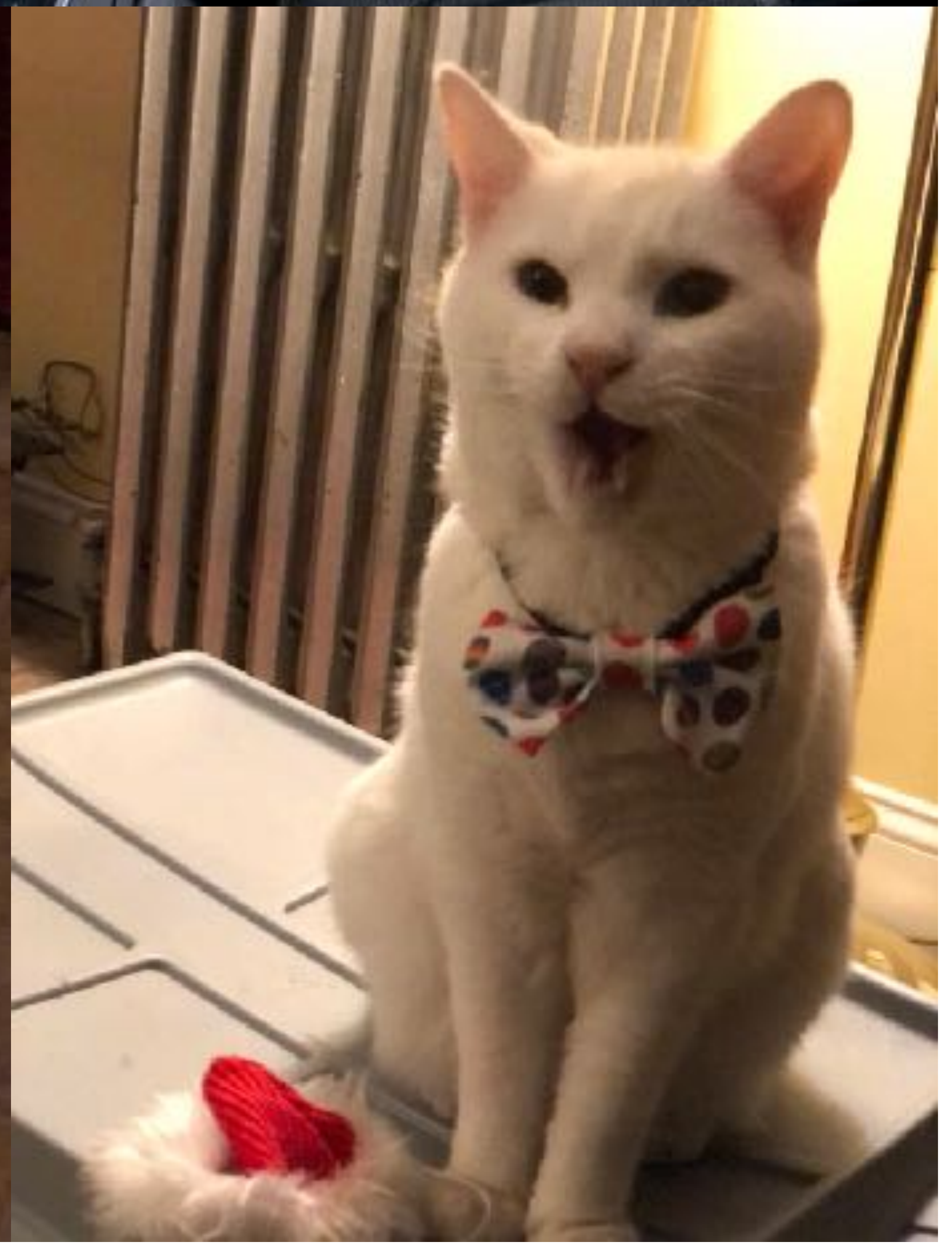


Thank you, MSEAS



Thank you, MSEAS

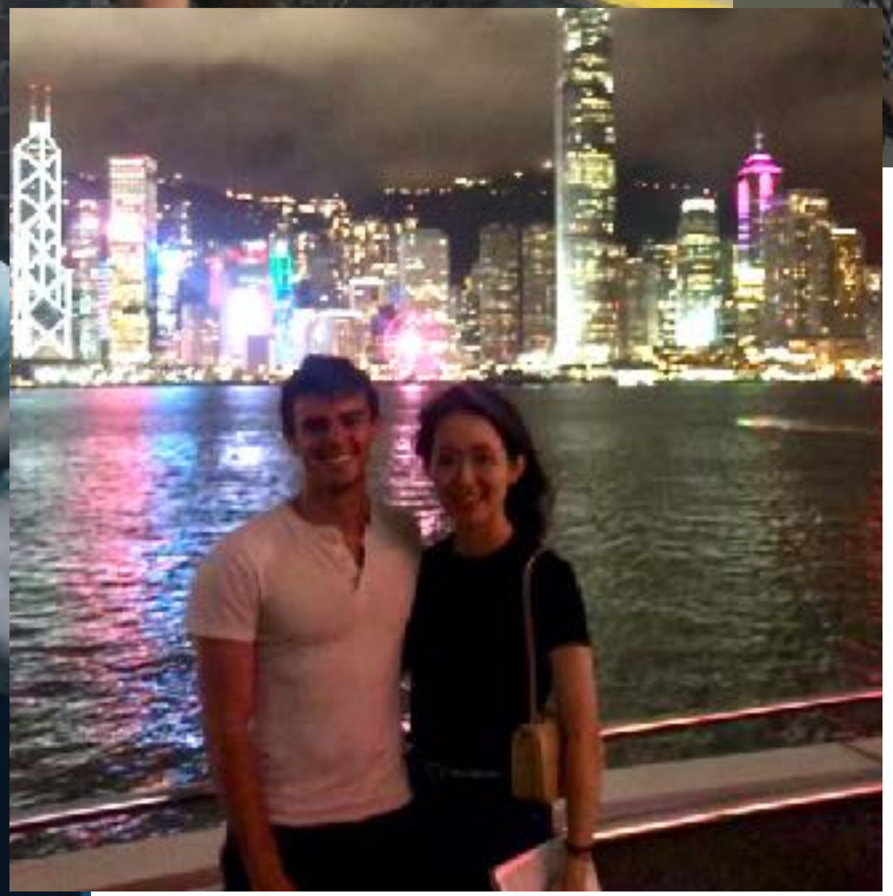




Thank you, Abhinav


















Thank you!

References (ocean modeling)

Cambon, G., Marchesiello, P., IRD, P. P., Debreu, L., Benshila, R., and Jullien, S. (2018). Croco User Guide. 49.

Fox-Kemper, B., Adcroft, A., Böning, C.W., Chassignet, E.P., Curchitser, E., Danabasoglu, G., Eden, C., England, M.H., Gerdes, R., Greatbatch, R.J. and Griffies, S.M., 2019. Challenges and prospects in ocean circulation models. *Frontiers in Marine Science*, *6*, p.65.

Mara Freilich and Amala Mahadevan. Coherent pathways for subduction from the surface mixed layer at ocean fronts. Journal of Geophysical Research: Oceans, 126(5):e2020JC017042, 2021

Hamlington, P. E., Van Roekel, L. P., Fox-Kemper, B., Julien, K., and Chini, G. P. (2014). Langmuir-submesoscale interactions: descriptive analysis of multiscale frontal spin-down simulations. J. Phys. Oceanogr. 44, 2249–2272. doi: 10.1175/JPO-D-13-0139.1

Jochen Kaempf. Wind-driven overturning, mixing and upwelling in shallow water: A nonhydrostatic modeling study. Journal of Marine Science and Engineering, 5(4):47, 2017.

Mayer, F.T. and Fringer, O.B., 2021. Resolving nonhydrostatic effects in oceanic lee waves. *Ocean Modelling*, *159*, p.101763.

Pan, W., Kramer, S.C. and Piggott, M.D., 2019. Multi-layer non-hydrostatic free surface modelling using the discontinuous Galerkin method. Ocean Modelling, 134, pp.68-83.

Pan, W., Kramer, S.C. and Piggott, M.D., 2021. A σ-coordinate non-hydrostatic discontinuous finite element coastal ocean model. Ocean Modelling, 157, p.101732.

Mahadevan, A. (2006). Modeling vertical motion at ocean fronts: are nonhydrostatic effects relevant at submesoscales? *Ocean Model.* 14, 222–240. doi: 10.1016/j.ocemod.2006.05.005

Ramadhan, A., Wagner, G., Hill, C., Campin, J.M., Churavy, V., Besard, T., Souza, A., Edelman, A., Ferrari, R. and Marshall, J., 2020. Oceananigans. jl: Fast and friendly geophysical fluid dynamics on GPUs. *Journal of Open Source Software*, *5*(53).

Suzuki, N., Fox-Kemper, B., Hamlington, P. E., and Van Roekel, L. P. (2016). Surface waves affect frontogenesis. J. Geophys. Res. Oceans 121, 1–28. doi: 10.1002/2015JC011563

Vitousek, Sean, and Oliver B. Fringer. "A nonhydrostatic, isopycnal-coordinate ocean model for internal waves." *Ocean Modelling* 83 (2014): 118-144.

References (DG methods)

N. Fehn, W.A. Wall ,and M. Kronbichler. On the stability of projection methods for the incompressible Navier–Stokes equations based on high-order discontinuous galerkin discretizations. Journal of Computational Physics, 351:392–421, 2017.

N. Fehn, W. A. Wall, and M. Kronbichler. Robust and efficient discontinuous galerkin methods for under-resolved turbulent incompressible flows. Journal of Computational Physics, 372:667–693, 2018.

N. Fehn, W.A. Wall, M. Kronbichler, Efficiency of high-performance discontinuous Galerkin spectral element methods for under-resolved turbulent incompressible flows, Int. J. Numer. Methods Fluids (2018) 1–23, https://doi.org/10.1002/fld.4511.

M. Kronbichler. The Discontinuous Galerkin Method: Derivation and Properties. In M. Kronbichler and P.-O. Persson, editors, Efficient High-Order Discretizations for Computational Fluid Dynamics, pages 1–55. Springer International Publishing, 2021.

Kronbichler, M., 2021. High-Performance Implementation of Discontinuous Galerkin Methods with Application in Fluid Flow. In *Efficient High-Order Discretizations for Computational Fluid Dynamics* (pp. 57-115). Springer, Cham.

References (RL/AMR)

C. Foucart, A. Charous, and P. FJ Lermusiaux. 2022. Deep reinforcement learning for adaptive mesh refinement. [sub judice]

Yang, J., Dzanic, T., Petersen, B., Kudo, J., Mittal, K., Tomov, V., Camier, J.S., Zhao, T., Zha, H., Kolev, T. and Anderson, R., 2021. Reinforcement learning for adaptive mesh refinement. arXiv preprint arXiv:2103.01342.

H. De Sterck, T. Manteuffel, S. McCormick, J. Nolting, J. Ruge, and L. Tang. Efficiency-based h-and hp-refinement strategies for finite element methods. Numerical Linear Algebra with Applications, 15(2-3):89–114, 2008.

T. Plewa, T. Linde, V. G. Weirs, et al. Adaptive mesh refinement-theory and applications. Springer, 2005.

A.Krishnapriyan, A.Gholami, S.Zhe, R.Kirby, and M.W.Mahoney. Characterizing possible failure modes in physics-informed neural networks. Advances in Neural Information Processing Systems, 34, 2021.

D. Arndt, W. Bangerth, B. Blais, M. Fehling, R. Gassmöller, T. Heister, L. Heltai, U. Köcher, M. Kronbichler, M. Maier, P. Munch, J.-P. Pelteret, S. Proell, K. Simon, B. Turcksin, D. Wells, and J. Zhang. The deal.II library, version 9.3. Journal of Numerical Mathematics, 29(3):171–186, 2021.

Extra slides

Free surface boundary conditions

8. Appendix A

Derivation of free-surface boundary conditions

The conceptual basis of the projection scheme is the idea that we can define a "total effective nonhydrostatic pressure" which has component due to the free-surface η and a component due to the nonhydrostatic pressure p'.

$$\mathrm{P}'_{\text{eff}} = p' + g\eta, \tag{46}$$

which explains the terms $\nabla \mathrm{P}' = \nabla p' + g\nabla\eta$ in the momentum equation (1). The free-surface corrector and intermediate projection step enforces the free-surface continuity equation and projects out the velocity divergence due to the free-surface pressure $g\eta$. The pressure-corrector step projects out the velocity divergence due to the nonhydrostatic pressure p'.

To enforce the free-surface evolution equation (3), we approximate $\boldsymbol{u}^{k+1}$ by $\bar{\bar{\boldsymbol{u}}}^{k+1} = \bar{\boldsymbol{u}}^{k+1} - a\Delta t\nabla_{xy}\left(g\delta\eta^{k+1}\right)$:

$$\begin{aligned}
&\frac{\eta^{k+1}}{a\Delta t} + \nabla\cdot\int_{-H}^{\eta}\left(\bar{\boldsymbol{u}}^{k+1} - a\Delta t\nabla_{xy}\left(g\delta\eta^{k+1}\right)\right)\,dz = \frac{\eta^{k}}{a\Delta t} \\
&\Rightarrow \frac{\delta\eta^{k+1}}{a\Delta t} - \nabla\cdot\left(a\Delta t g\left[\eta^{k}+H\right]\nabla\delta\eta^{k+1}\right) = -\nabla\cdot\int_{-H}^{\eta}\bar{\boldsymbol{u}}^{k+1}\,dz.
\end{aligned} \tag{47}$$

For a Dirichlet boundary condition on the velocity $\boldsymbol{u}^{k+1}$, making use of the definition of $\bar{\bar{\boldsymbol{u}}}^{k+1}$, we have

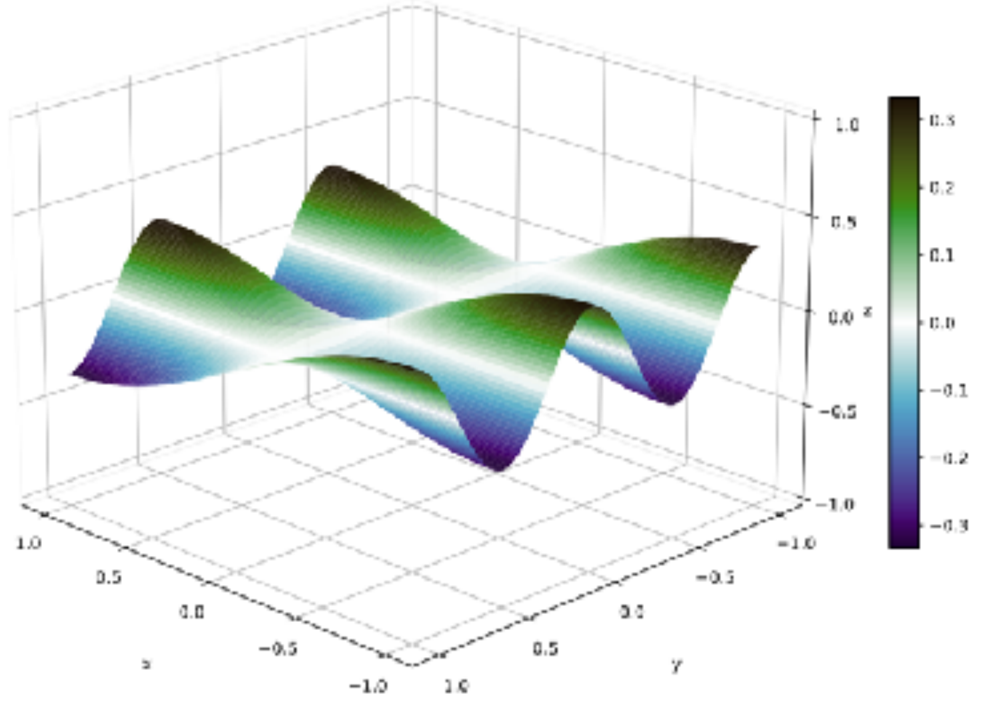
$$\begin{aligned}
&\nabla\delta\eta^{k+1} = \frac{\bar{\boldsymbol{u}}^{k+1} - \bar{\bar{\boldsymbol{u}}}^{k+1}}{a\Delta t g} \\
&\Rightarrow g_{N,\delta\eta} = \nabla\delta\eta^{k+1}\cdot\boldsymbol{n} = \frac{1}{a\Delta t g}\left(\bar{\boldsymbol{u}}^{k+1} - \boldsymbol{g}_{D,\boldsymbol{u}}\right)\cdot\boldsymbol{n}
\end{aligned} \tag{48}$$

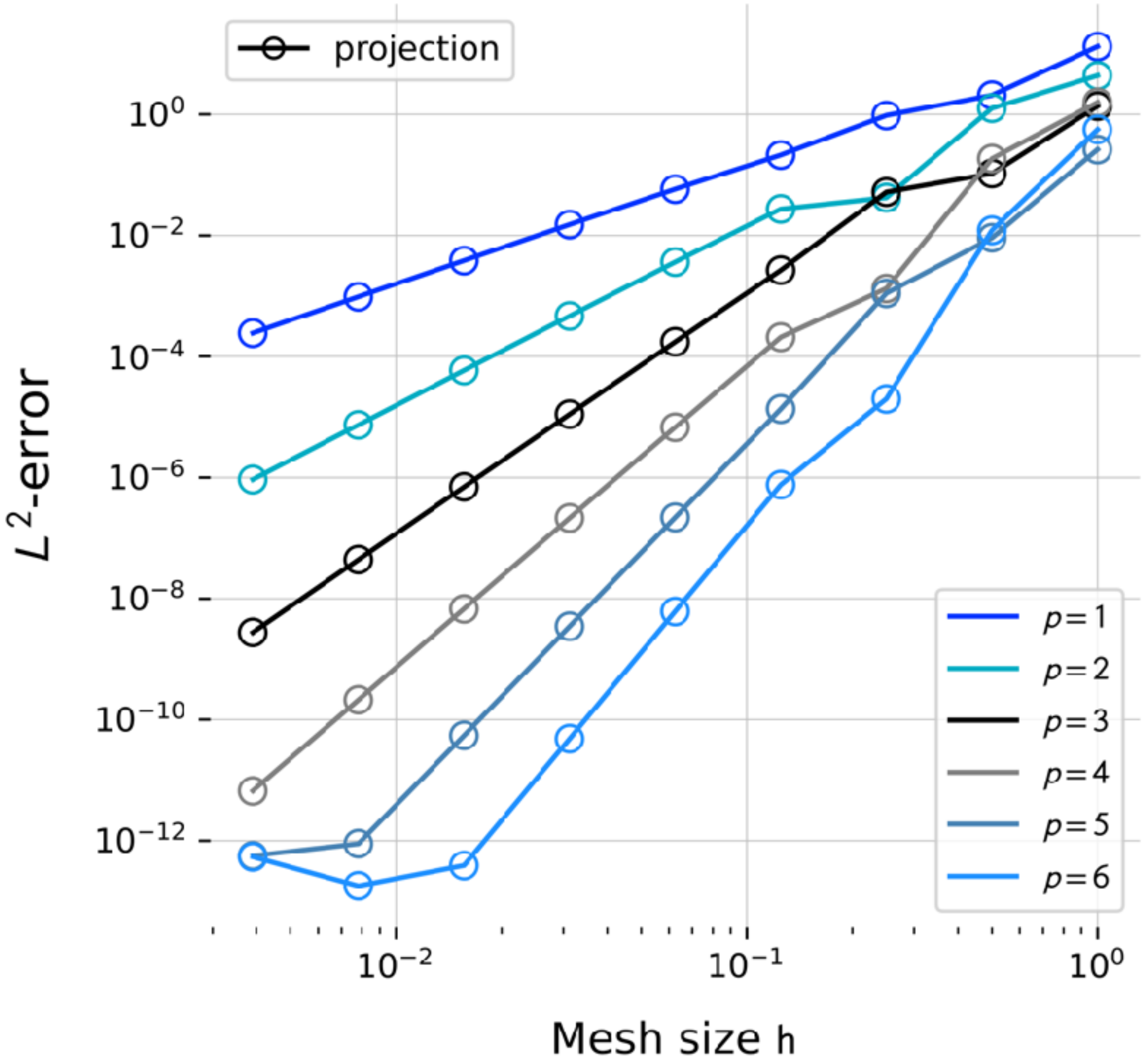
which enforces that the horizontal component of the second intermediate velocity $\bar{\bar{\boldsymbol{u}}}^{k+1}$ has the value $\boldsymbol{g}_{D,\boldsymbol{u}}$ on the boundary Γ_D after the intermediate updates[4]. In the case where we were able to enforce that $\bar{\boldsymbol{u}}^{k+1} = \boldsymbol{g}_{D,\boldsymbol{u}}$ exactly, this reduces to a zero-Neumann condition; however, since HDG methods may only impose boundary conditions weakly through the edge-space M_h^p, the value $g_{N,\delta\eta}$ may not be exactly zero numerically[5].

Pure Neumann problem: convergence

Singular system

- Arises in pressure-poisson equation
- Rank-one deficient
- Several approaches to treat singularity [2]:
  - point constraint
  - penalty method
  - subspace projection
- All converge optimally (but at different costs), no rigorous comparison has ever been done [1]



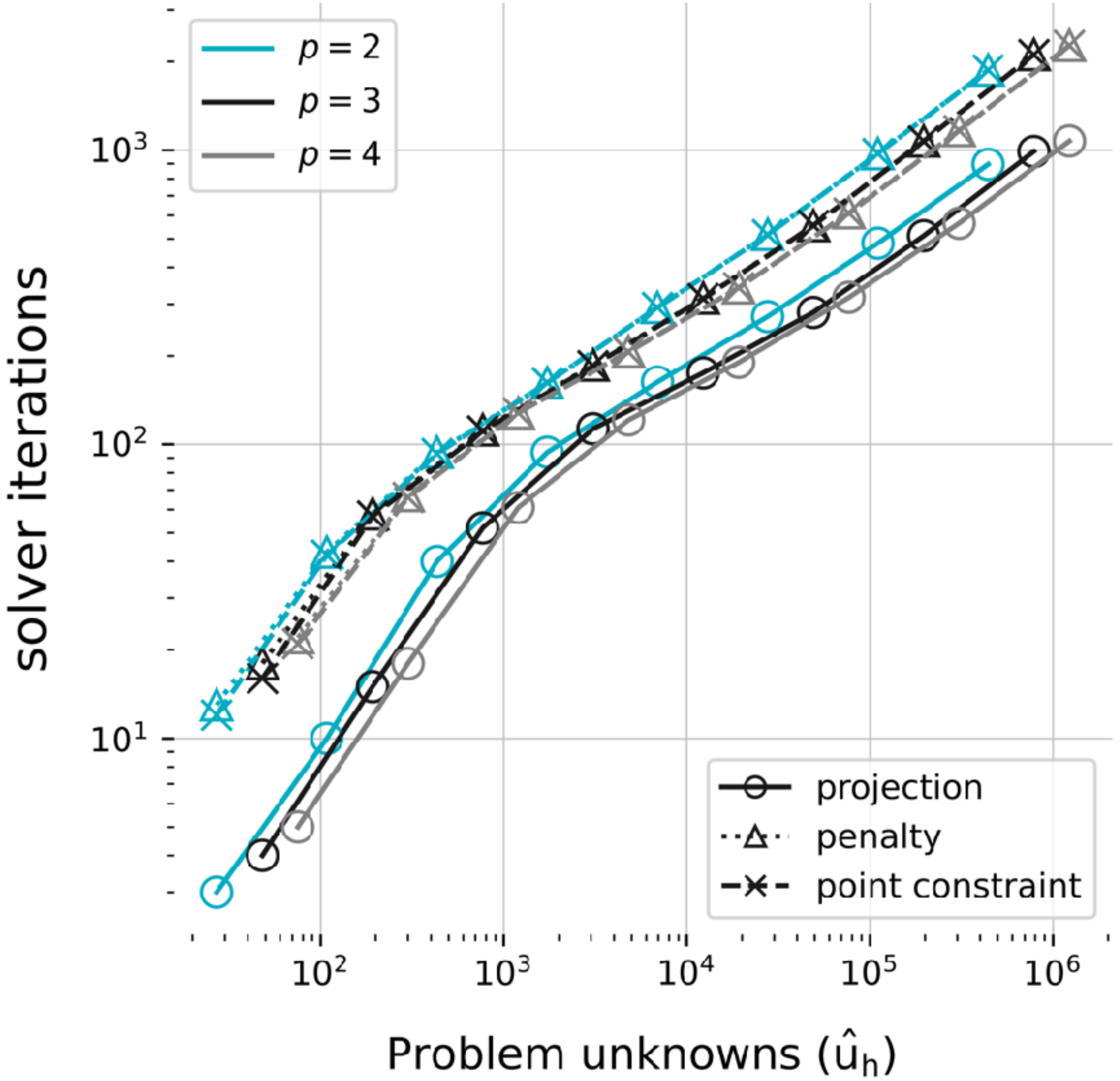


Pure Neumann problem: benchmarking

Singular system

- Several approaches to treat singularity:
  - point constraint
  - penalty method
  - subspace projection

Compare their performance in terms of iterative solver iterations



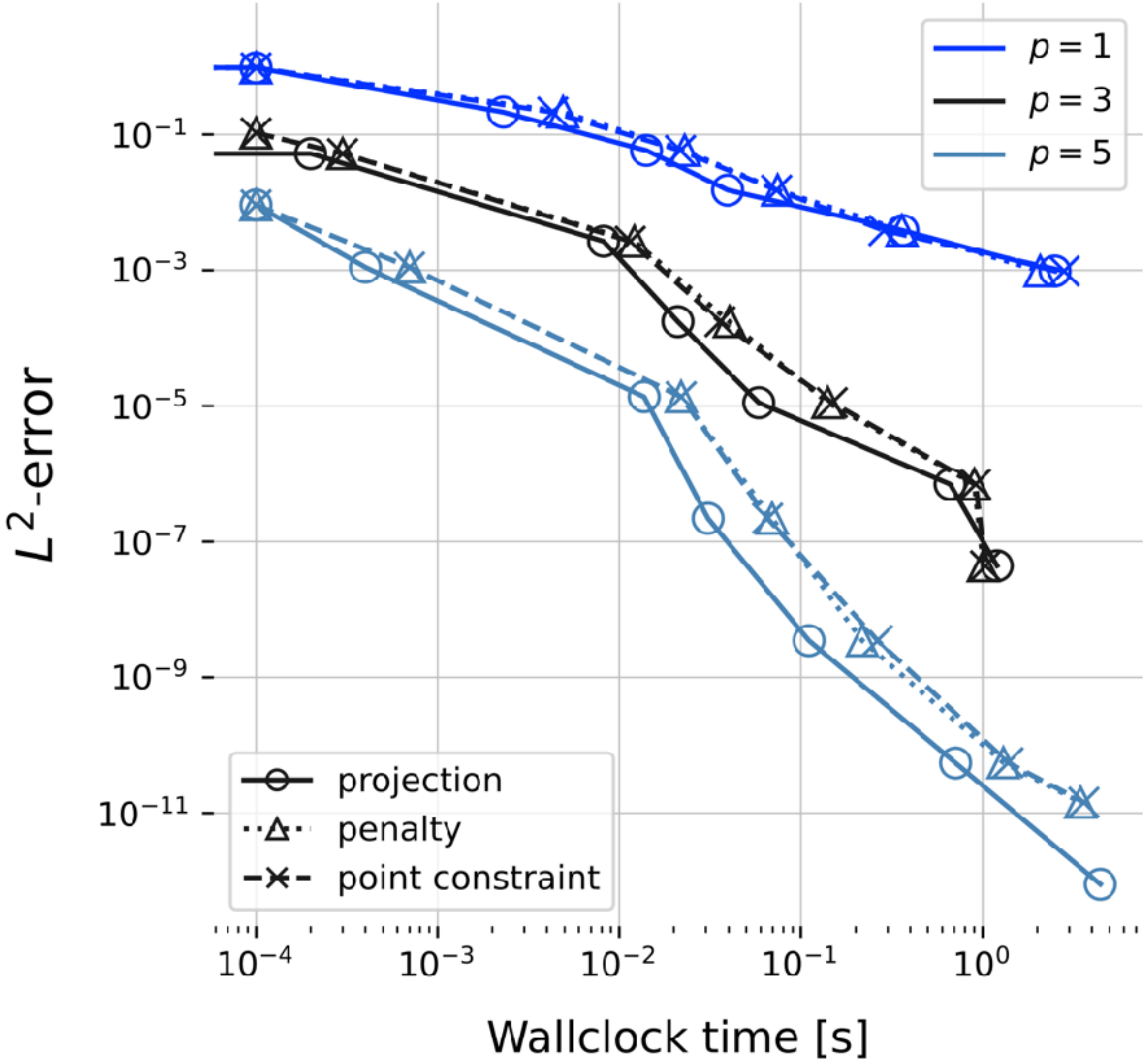
Pure Neumann problem: benchmarking

Singular system

- Several approaches to treat singularity:
  - point constraint
  - penalty method
  - subspace projection

Solver iterations can be misleading due to the cost of applying each iteration; measure error as a function of wall-clock time to solution.

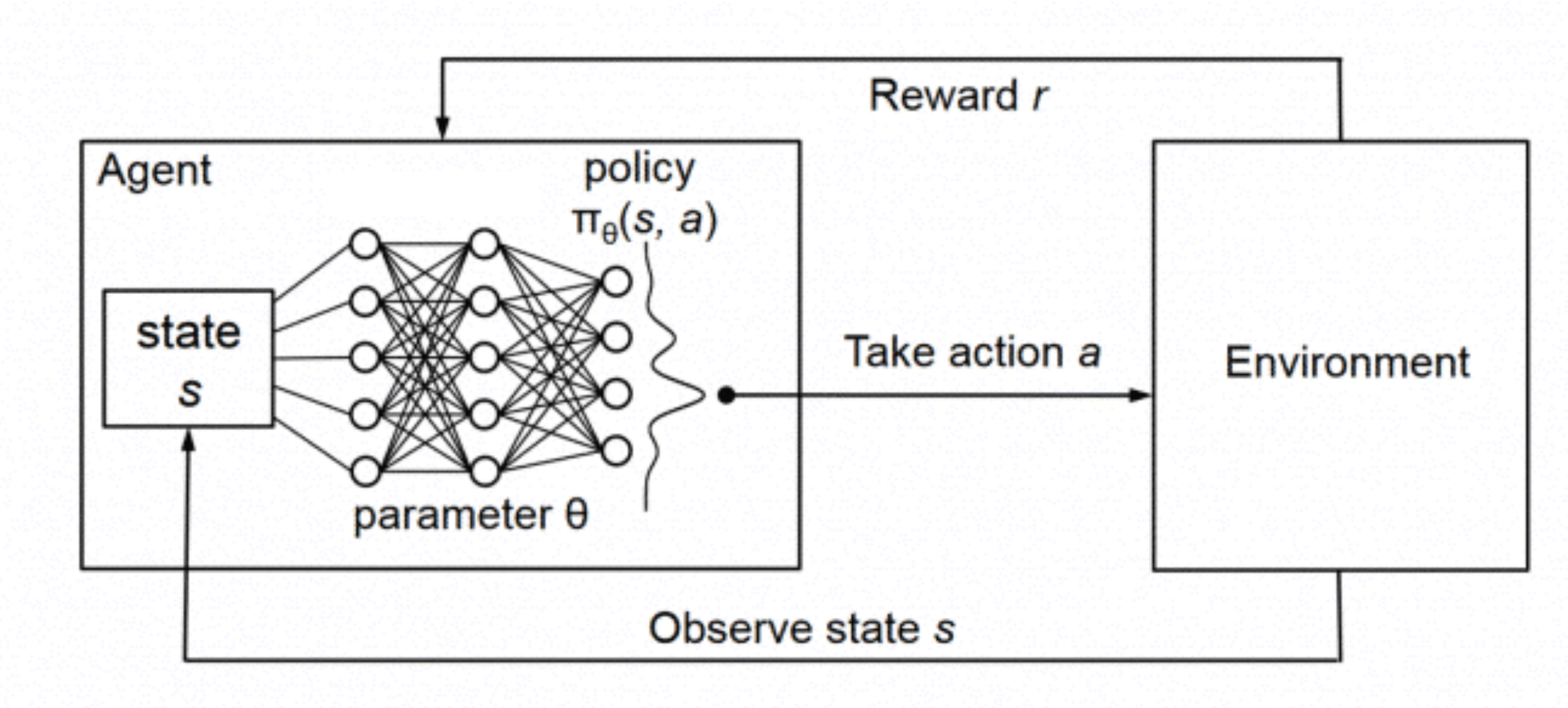
Subspace projection is unambiguously the best, especially at high-order

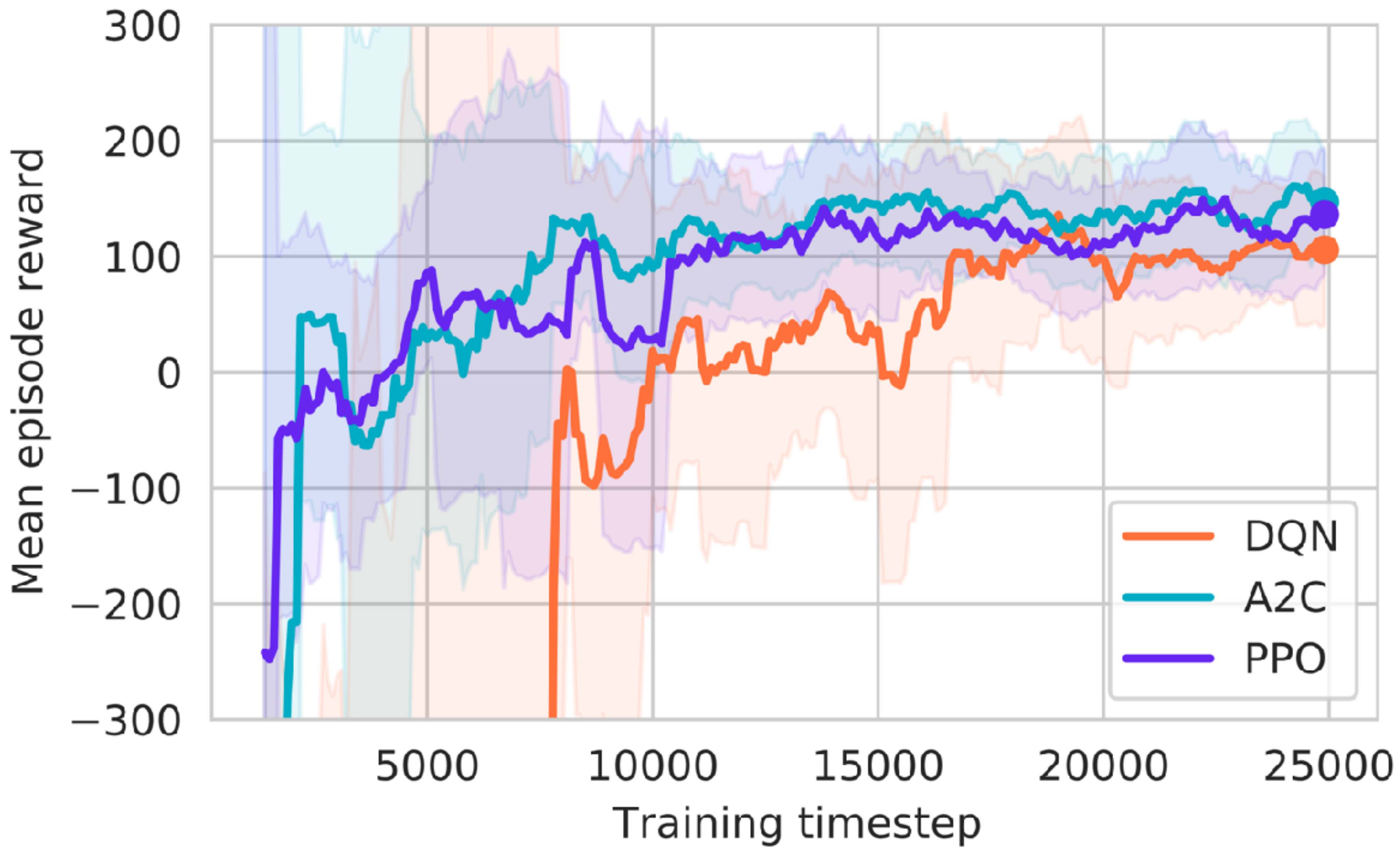


Deep reinforcement learning: training and deployment

"deep" RL:

- represent value function as neural network
- use each reward signal (experience) to update network during **training**
- once trained, agent is "**deployed**" by querying policy network for recommendation based on current state and action
- neural networks are universal function approximators (Kornik et al., 1989) and can learn arbitrarily complex policies





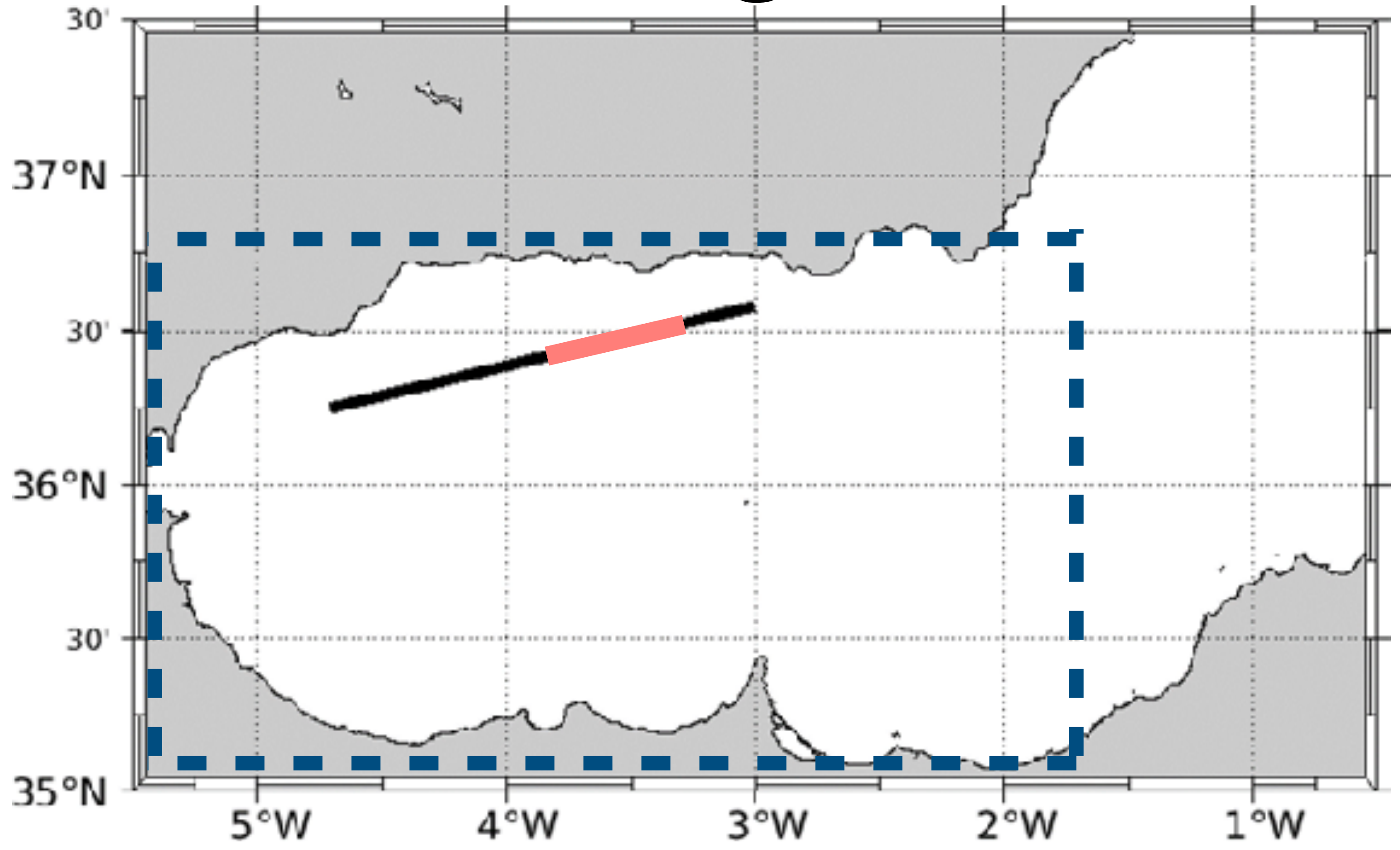
(Foucart et al., JCP-2022-sub)

Application: 2D model nesting within a large hydrostatic code (MSEAS-PE)

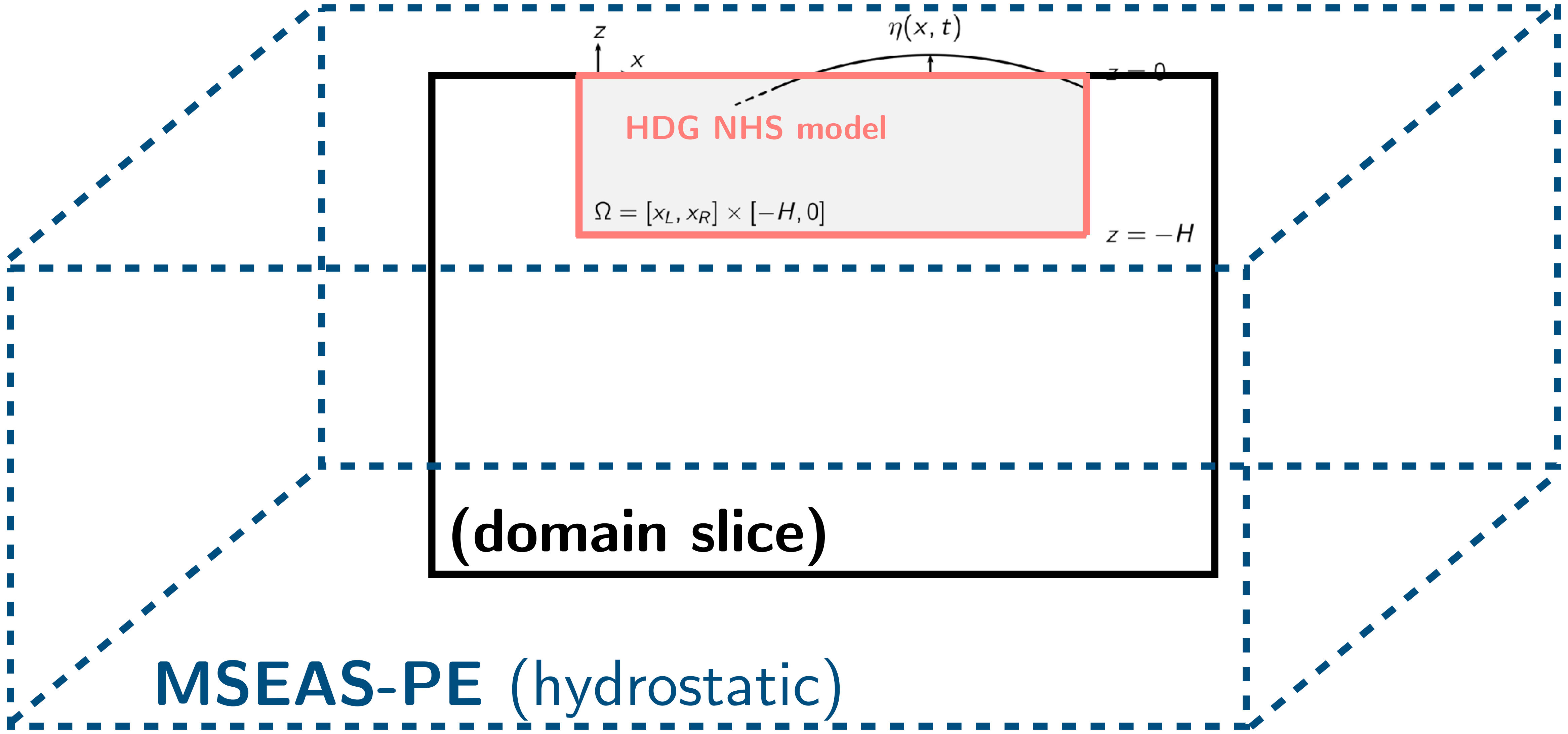
Region of investigation:

- **Black line:** denotes a section in the Alboran Sea along which **wind-driven instabilities were observed** in the hydrostatic MSEAS PE simulations
- The section starts in the west along the northern edge of the West Alboran Gyre and extends to the east-by-northeast out of the gyre towards the Spanish coast.
- Implemented 2D HDG NHS model nesting and initialization from real data
- Goal: model instabilities in the mixed layer and compare to HS simulation output

modeling domains



(not to scale)



NHS model: boundary conditions

	top	bottom	sides
$\bar{u}$	Γ_D	Γ_D	Γ_D
$\bar{w}$	Γ_N	Γ_D	Γ_D
$\delta\eta$	-	-	Γ_N
$\delta p'$	Γ_D	Γ_N	Γ_N
ρ'	Γ_D	Γ_D	Γ_D

[1] Foucart et al., Ocean Modelling. in prep. 2023a.