THE CYPRUS INSTITUTE
RESEARCH • TECHNOLOGY • INNOVATION

AI modeling for wetting hydrodynamics

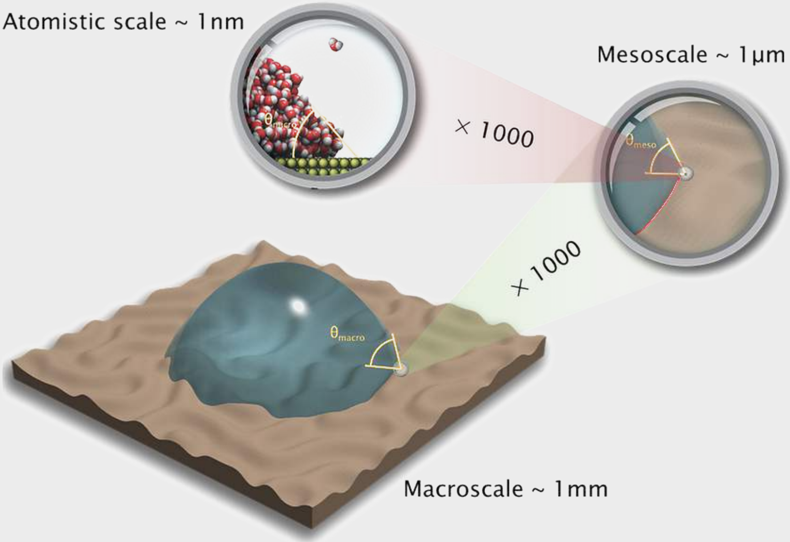
Andreas Demou, Nikos Savva
The Cyprus Institute

Aqtivate school, 29 Feb. 2024

1. Intro to wetting hydrodynamics
2. Intro to the FNO
3. Using low-fidelity data
4. High-fidelity data: gravity-driven
5. High-fidelity data: capillary-driven

1. Intro to wetting hydrodynamics

Wetting hydrodynamics

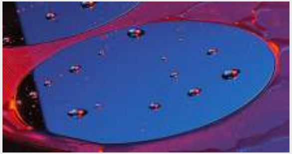


Wetting hydrodynamics - applications

Water harvesting

Tribology

Self-cleaning

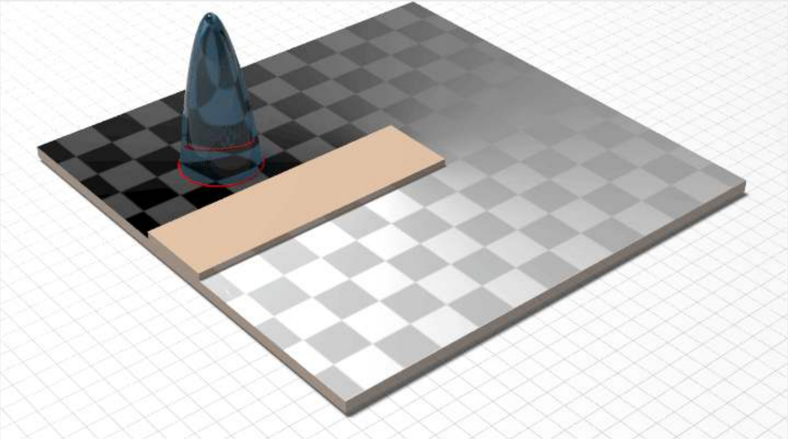


Wetting hydrodynamics - Microfluidic lab-on-a-chip

(MIT Media Lab)

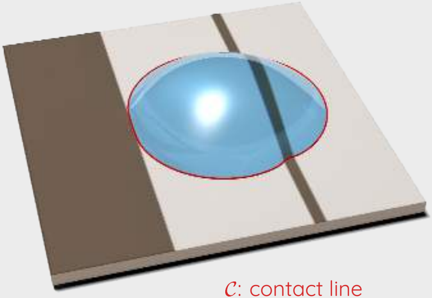
Numerical simulations

Droplet of constant volume migrating from less hydrophilic (darker-shaded) to more hydrophilic (lighter-shaded) regions



Overview

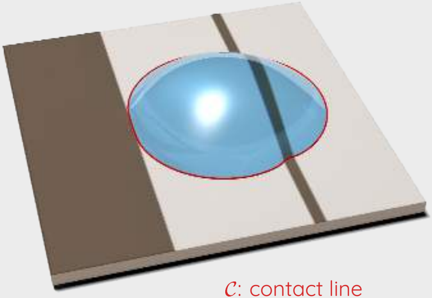
Simulation Data → Evolution of Contact Line



$\mathcal{C}$: contact line

Overview

Simulation Data → Evolution of Contact Line → (data-driven modelling) → Surrogate low-order models

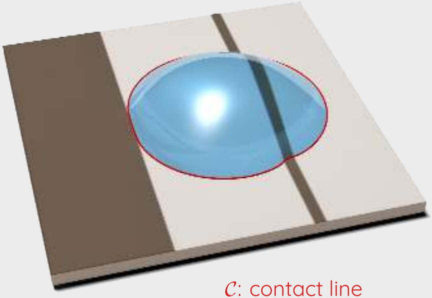


$\mathcal{C}$: contact line

Overview

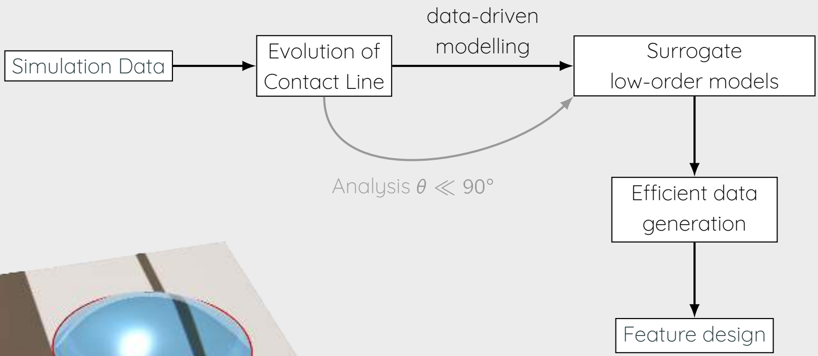
Simulation Data → Evolution of Contact Line $\xrightarrow{\text{data-driven modelling}}$ Surrogate low-order models

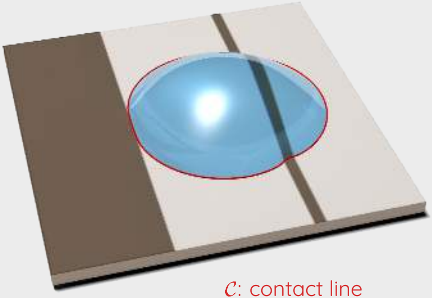
Evolution of Contact Line $\xrightarrow{\text{Analysis } \theta \ll 90^\circ}$ Surrogate low-order models



$\mathcal{C}$: contact line

Overview





$\mathcal{C}$: contact line

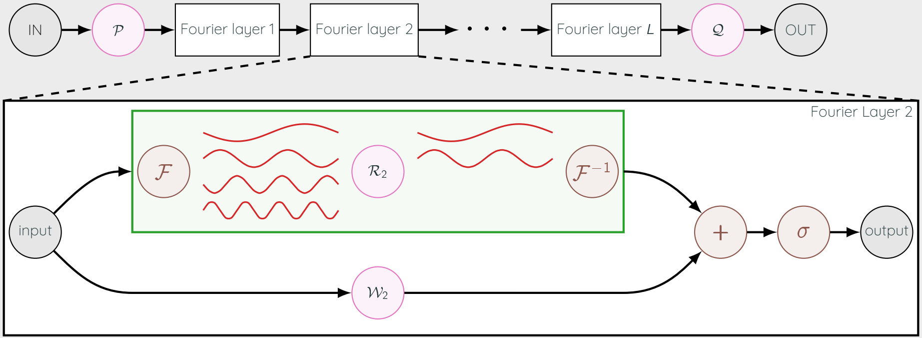
2. Intro to the FNO

AI modelling - Fourier Neural Operator (FNO)

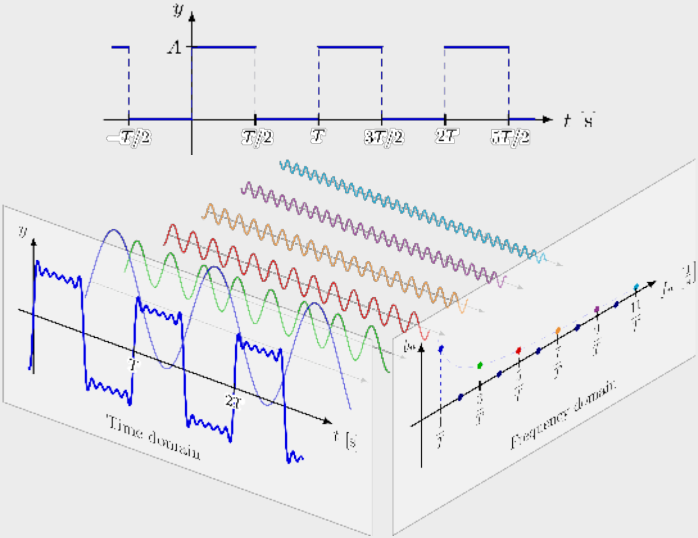
Learn contact line dynamics in a data-driven manner, by considering the mapping:

$$G = (\text{aux. data}) \to \{\text{Solution}\}$$

Key idea: A neural operator can approximate G through the Fourier space.



Fourier decomposition (1D)



AI modelling - FNO Python implementation

```python
def forward(self, x):
    # apply lifting operator
    x = self.P(x)

    # Fourier layer (1)
    x1 = self.conv1(x)
    x2 = self.w1(x)
    x  = x1 + x2
    x  = F.gelu(x)

    ...

    # Fourier layer (L)
    x1 = self.convL(x)
    x2 = self.wL(x)
    x  = x1 + x2
    x  = F.gelu(x)

    # apply projection operator
    x = self.Q(x)

    return x
```

AI modelling - FNO Python implementation

```
def forward(self, x):
    # apply lifting operator
    x = self.P(x)

    # Fourier layer (1)
    x1 = self.conv1(x)
    x2 = self.w1(x)
    x  = x1 + x2
    x  = F.gelu(x)

    ...

    # Fourier layer (L)
    x1 = self.convL(x)
    x2 = self.wL(x)
    x  = x1 + x2
    x  = F.gelu(x)

    # apply projection operator
    x = self.Q(x)

    return x
```

```
class SpectralConv1d(nn.Module):
    def __init__(self, in_channels, out_channels, modes1):
        ...

    def compl_mul1d(self, input, weights):
        return torch.einsum("bix,iox->box", input, weights)

    def forward(self, x):
        batchsize = x.shape[0]

        #Compute Fourier coefficients
        x_ft = torch.fft.rfft(x)

        # Multiply relevant Fourier modes
        out_ft = torch.zeros(...)
        out_ft[:,:,:self.modes1] = self.compl_mul1d(x_ft[:,:,:self.modes1],\
                                                     self.weights1)
        #Return to physical space
        x = torch.fft.irfft(out_ft, n=x.size(-1))

        return x
```

3. Using low-fidelity data

Governing Equations: Long-wave approximation

Assumptions

- strong surface tension
- negligible inertial effects
- small contact angles

Non-dimensional governing PDE

$$\partial_t h + \boldsymbol{\nabla} \cdot \left[h(h^2 + \lambda^2) \boldsymbol{\nabla} \nabla^2 h \right] = 0$$

Boundary conditions along the contact line $\mathcal{C}$ (ν is the unit outward normal on $\mathcal{C}$)

Thickness vanishes: $h|_{\mathcal{C}} = 0$

Contact angle: $\tan\theta|_{\mathcal{C}} = |\boldsymbol{\nabla} h|_{\mathcal{C}} = -h_\nu = \vartheta^*$

Kinematic BC: $\left(\partial_t \boldsymbol{c} - \lambda^2 \boldsymbol{\nabla} \nabla^2 h\big|_{\mathcal{C}}\right) \cdot \boldsymbol{\nu} = 0$

Approach A: Fully data-driven solution

Contact line $c(t_i)$ is discretised with 128 points and time t_i is discretised uniformly.

1st iteration

Input: $\{c(t_1), c(t_2), \ldots, c(t_{10}), \vartheta_c^*(t_1), \vartheta_c^*(t_2), \ldots, \vartheta_c^*(t_{10})\}$

Output: $c(t_{11})$, i.e. subsequent solution

Approach A: Fully data-driven solution

Contact line $c(t_i)$ is discretised with 128 points and time t_i is discretised uniformly.

1st iteration

Input: $\{c(t_1), c(t_2), \ldots, c(t_{10}), \vartheta_c^*(t_1), \vartheta_c^*(t_2), \ldots, \vartheta_c^*(t_{10})\}$

Output: $c(t_{11})$, i.e. subsequent solution

2nd iteration

Input: $\{c(t_2), c(t_3), \ldots, c(t_{11}), \vartheta_c^*(t_2), \vartheta_c^*(t_3), \ldots, \vartheta_c^*(t_{11})\}$

Output: $c(t_{12})$

...

This procedure is applied iteratively to get the solution up to t_{fin}.

Approach A: Fully data-driven solution

Contact line $c(t_i)$ is discretised with 128 points and time t_i is discretised uniformly.

1st iteration

Input: $\{c(t_1), c(t_2), \ldots, c(t_{10}), \vartheta_c^*(t_1), \vartheta_c^*(t_2), \ldots, \vartheta_c^*(t_{10})\}$

Output: $c(t_{11})$, i.e. subsequent solution

2nd iteration

Input: $\{c(t_2), c(t_3), \ldots, c(t_{11}), \vartheta_c^*(t_2), \vartheta_c^*(t_3), \ldots, \vartheta_c^*(t_{11})\}$

Output: $c(t_{12})$

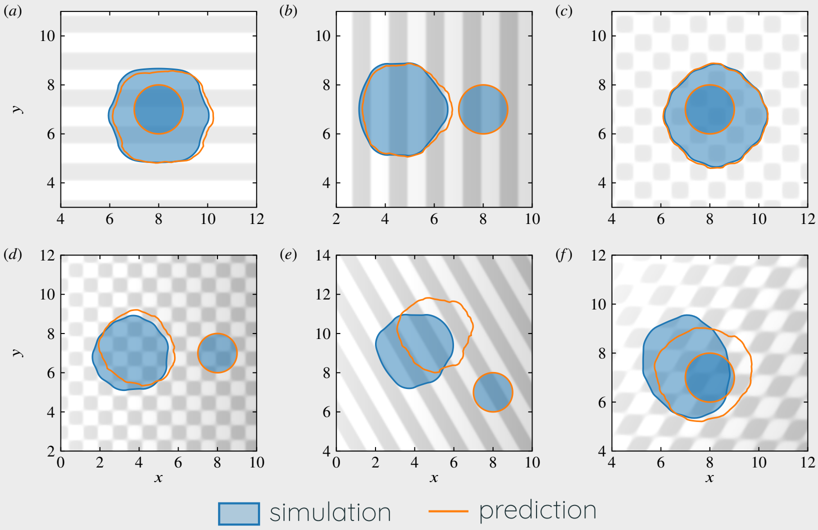
...

This procedure is applied iteratively to get the solution up to t_{fin}.

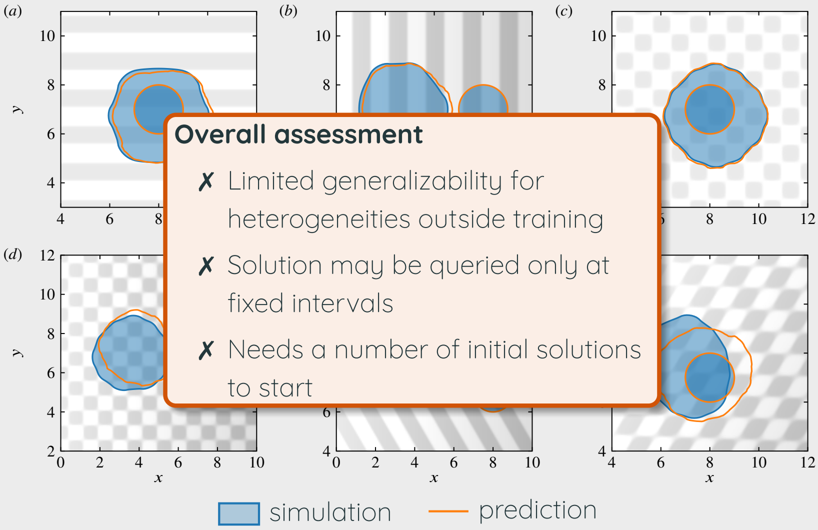
Cumulative error:

$$\frac{1}{N_{train}} \sum_{n=1}^{N_{train}} \frac{1}{i_{fin} - 10} \sum_{i=11}^{i_{fin}} \frac{\left\| \boldsymbol{c}_{AI}^n(t_i) - \boldsymbol{c}_{ref}^n(t_i) \right\|_2}{\left\| \boldsymbol{c}_{ref}^n(t_i) \right\|_2}$$

Approach A: Testing



Approach A: Testing

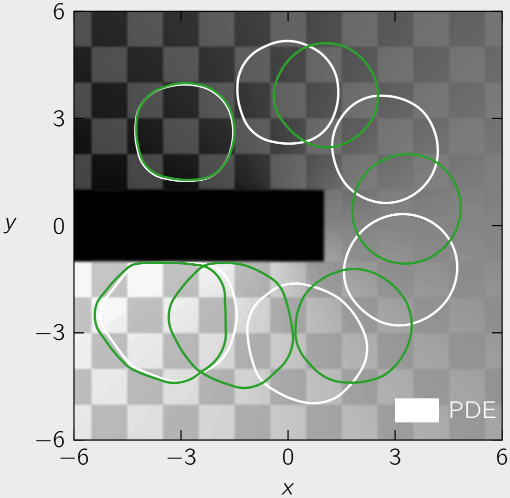


Overall assessment

- ✗ Limited generalizability for heterogeneities outside training
- ✗ Solution may be queried only at fixed intervals
- ✗ Needs a number of initial solutions to start

Inspiration: Matched asymptotic analysis

Savva & Groves (PRF 2021)



Compare reduced model with PDE solutions:

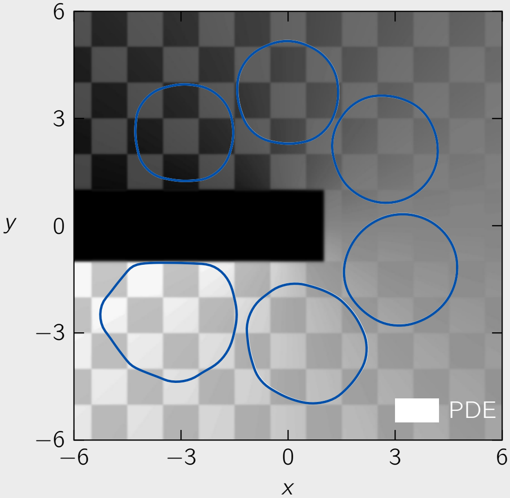
$$\boldsymbol{c} = (x_c + a\cos\phi, y_c + a\sin\phi) \text{ with } a = \sum_{m=0}^{M} a_m(t)\mathrm{e}^{im\phi}$$

$$\frac{\theta^3 - \vartheta^{*3}}{3} = \partial_t \boldsymbol{c}\cdot\boldsymbol{\nu}|\ln\lambda| + \partial_t \boldsymbol{c}\cdot\boldsymbol{\nu}\ln(ea\vartheta^*) - \sum_{m=0}^{M}\Big[\beta_m U_m - m\tilde{\beta}_m^0 \frac{a_m}{a_0}U_0 + \frac{m-1}{2a_0}(\gamma_m - \tilde{\beta}_m^-)a_{m-1}U_1 - \frac{m+1}{2a_0}(2\beta_m - \gamma_m + \tilde{\beta}_m^+)a_{m+1}U_1^*\Big]\mathrm{e}^{im\phi},$$

with $U_m = \dot{a}_m$, $m \neq 1$ and $U_1 = \dot{x}_c - i\dot{y}_c$

Inspiration: Matched asymptotic analysis

Savva & Groves (PRF 2021)



Compare reduced model with PDE solutions:

$$\boldsymbol{c} = (x_c + a\cos\phi, y_c + a\sin\phi) \text{ with } a = \sum_{m=0}^{M} a_m(t)\mathrm{e}^{im\phi}$$

$$\frac{\theta^3 - \vartheta^{*3}}{3} = \partial_t \boldsymbol{c}\cdot\boldsymbol{\nu}|\ln\lambda| + \partial_t \boldsymbol{c}\cdot\boldsymbol{\nu}\ln(\mathrm{e}a\vartheta^*)$$
$$-\sum_{m=0}^{M}\Big[\beta_m U_m - m\tilde{\beta}_m^0 \frac{a_m}{a_0}U_0 + \frac{m-1}{2a_0}(\gamma_m - \tilde{\beta}_m^-)a_{m-1}U_1$$
$$-\frac{m+1}{2a_0}(2\beta_m - \gamma_m + \tilde{\beta}_m^+)a_{m+1}U_1^*\Big]\,\mathrm{e}^{im\phi},$$

with $U_m = \dot{a}_m$, $m \neq 1$ and $U_1 = \dot{x}_c - i\dot{y}_c$

Approach B: AI-assisted modelling

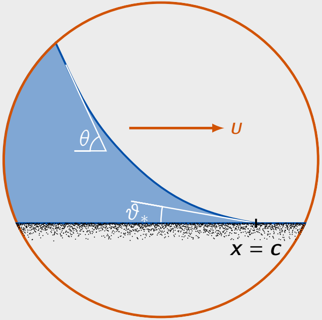
The data-driven model corrects an analytically derived imperfect model, in the spirit of Wan & Sapsis (JFM 2018).

Approach B: AI-assisted modelling

The data-driven model corrects an analytically derived imperfect model, in the spirit of Wan & Sapsis (JFM 2018).

Droplet velocity normal to the contact line, u_ν,

$$u_\nu = \bar{u}_\nu + G\left(\boldsymbol{c}, \bar{u}_\nu\right) \quad \text{with} \quad \bar{u}_\nu = \frac{\theta^3 - \vartheta^{*3}}{3\left|\ln\lambda\right|}$$



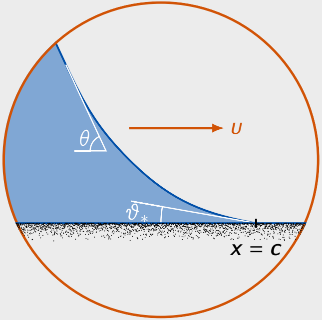
- θ and ϑ^* are the apparent and local contact angles
- λ the slip length
- $G\left(\boldsymbol{r}, u_\nu\right)$ higher-order terms, weak dependence on ϑ^*.

Input: $\{\boldsymbol{c}(t_i), \bar{u}_\nu(t_i)\}$

Output: $G(\boldsymbol{c}, \bar{u}_\nu)$

Approach B: AI-assisted modelling

The data-driven model corrects an analytically derived imperfect model, in the spirit of Wan & Sapsis (JFM 2018).



Droplet velocity normal to the contact line, u_ν,

$$u_\nu = \bar{u}_\nu + G\left(\boldsymbol{c}, \bar{u}_\nu\right) \quad \text{with} \quad \bar{u}_\nu = \frac{\theta^3 - \vartheta^{*3}}{3\,|\ln\lambda|}$$

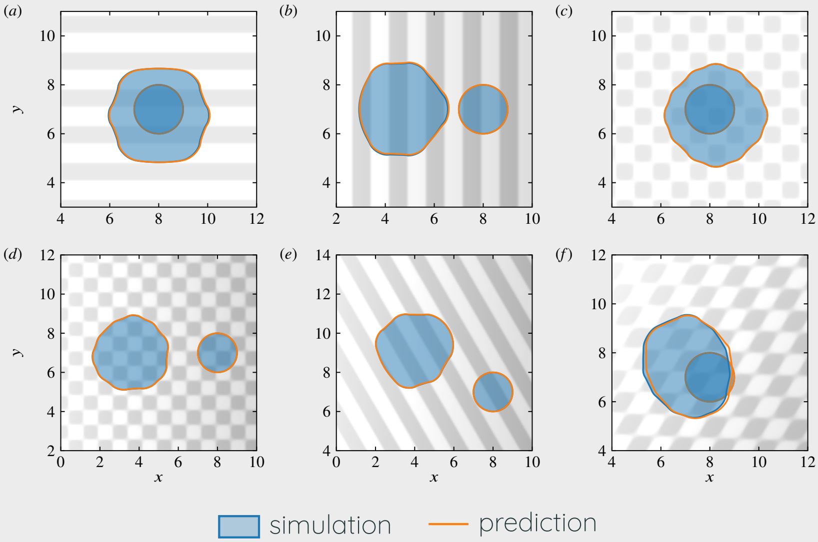
- θ and ϑ^* are the apparent and local contact angles
- λ the slip length
- $G\left(\boldsymbol{r}, u_\nu\right)$ higher-order terms, weak dependence on ϑ^*.

data-driven, implicit in t

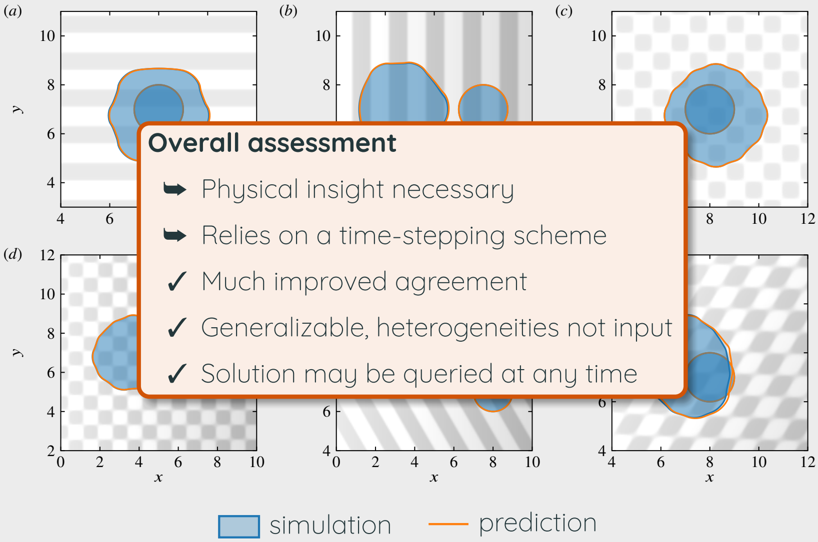
Input: $\{\boldsymbol{c}(t_i), \bar{u}_\nu(t_i)\}$

Output: $G(\boldsymbol{c}, \bar{u}_\nu)$

AI-assisted approach - Tests



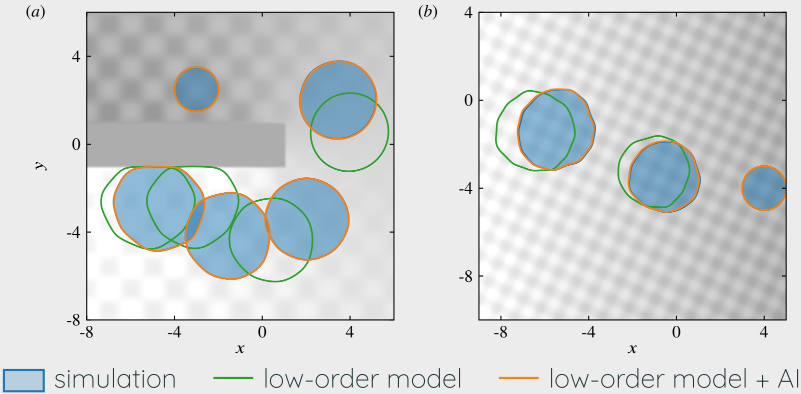
AI-assisted approach - Tests



Overall assessment

- ➥ Physical insight necessary
- ➥ Relies on a time-stepping scheme
- ✓ Much improved agreement
- ✓ Generalizable, heterogeneities not input
- ✓ Solution may be queried at any time

AI-assisted approach - Out of distribution



Published in **Data-centric Engineering** (Cambridge Univ. Press)

4. High-fidelity data: gravity-driven

Governing Equations:

Two-phase Navier-Stokes equations

$$\vec{\nabla} \cdot \vec{u} = 0,$$

$$\rho \left[\frac{\partial \vec{u}}{\partial t} + \vec{\nabla} \cdot (\vec{u}\vec{u}) \right] = -\vec{\nabla} p + \vec{\nabla} \cdot \left[\mu \left(\vec{\nabla} \vec{u} + \vec{\nabla} \vec{u}^T \right) \right] + \sigma \kappa \delta_\Gamma \vec{n} + \hat{\rho} \vec{g},$$

$$\frac{\partial C}{\partial t} + \vec{\nabla} \cdot (\vec{u} C) = 0, \text{ where } C(\vec{x}, t) = \begin{cases} 1 & \text{if } \vec{x} \in \text{liquid}, \\ 0 & \text{if } \vec{x} \in \text{gas}. \end{cases}$$

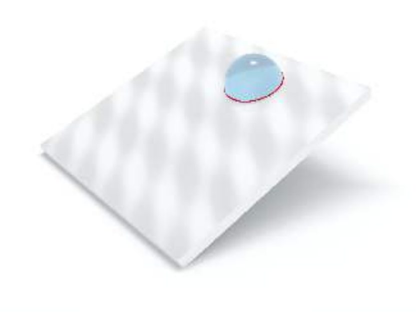
Physical properties ξ calculation: $\xi(\vec{x}, t) = \xi_1 C(\vec{x}, t) + \xi_2 (1 - C(\vec{x}, t))$.

Boundary conditions: impose local contact angle (chemical heterogeneity) on surface.

Data generation - Direct Numerical Simulations

Code: Basilisk

- random heterogeneities, from a 6-parameter functional form
- 20–90 dimensionless times, snapshot saved every 0.1 time units
- adaptive mesh refinement, local grid size between $1/2^5 - 1/2^8$



Dataset:

- 200 DNS cases
- 80,000 contact line snapshots

Reduced-order model

Cox (J. Fluid Mech. 1986)

Analysis near the contact line reveals

$$u_\nu^{COX} = \frac{\sigma}{\mu}\left(\frac{F(\vartheta_*) - F(\theta)}{\ln\left(\frac{\lambda}{r_0}\right) + \frac{Q_o}{f(\theta)} - \frac{Q_i}{f(\vartheta_*)}}\right)$$

- λ, slip length, scales with Δx
- σ surface tension; μ viscosity
- Q_o and Q_i are unspecified
- F and f are known

Reduced-order model

Cox (J. Fluid Mech. 1986)

Analysis near the contact line reveals

$$u_\nu^{COX} = \frac{\sigma}{\mu}\left(\frac{F(\vartheta_*) - F(\theta)}{\ln\left(\frac{\lambda}{r_0}\right) + \frac{Q_o}{f(\theta)} - \frac{Q_i}{f(\vartheta_*)}}\right)$$

- λ, slip length, scales with Δx
- σ surface tension; μ viscosity
- Q_o and Q_i are unspecified
- F and f are known

Obtaining θ assuming quasi-static dynamics

Given $\boldsymbol{c}$, obtain θ from the slope of the solution to the Young–Laplace eqn

$$-\sigma\nabla\cdot\hat{\boldsymbol{n}} = \Delta p, \qquad \hat{\boldsymbol{n}} \text{ the surface unit normal}$$

Δp is constant specified by the volume constraint.

➥ An approximate model for the solution of this equation was analystically derived, considering weakly deformed contact lines.

Cox (J. Fluid Mech. 1986)

Analysis near the contact line reveals

$$u_\nu^{COX} = \frac{\sigma}{\mu}\left(\frac{F(\vartheta_*) - F(\theta)}{\ln\left(\frac{\lambda}{r_0}\right) + \frac{Q_o}{f(\theta)} - \frac{Q_i}{f(\vartheta_*)}}\right)$$

- λ, slip length, scales with Δx
- σ surface tension; μ viscosity
- Q_o and Q_i are unspecified
- F and f are known

Cox's model does not include gravity effects

AI Approach: Train a model $\hat{u}_c$ to incorporate gravity in the simulations

$$u_c^{DNS} \approx u_c^{COX} + \hat{u}_c$$

AI model for introducing gravity effects

Net transport captured by the first harmonic of the contact line velocity

Input: snapshots of first harmonics of θ and ϑ_*, including Bo, a_i

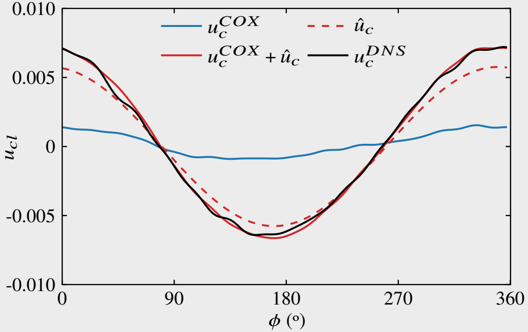
Output: snapshots of first harmonics of $u_c^{DNS} - u_c^{COX} \rightarrow \hat{u}_c$

AI model for introducing gravity effects

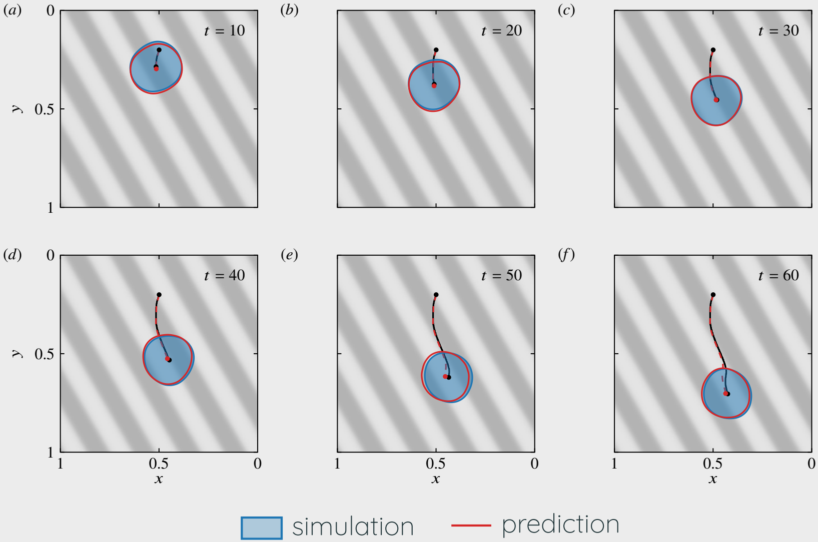
Net transport captured by the first harmonic of the contact line velocity

Input: snapshots of first harmonics of θ and ϑ_*, including Bo, a_i

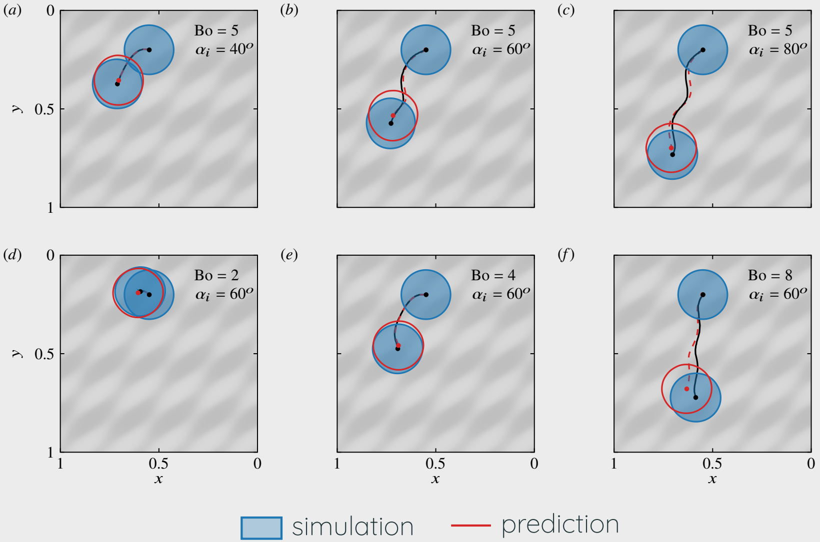
Output: snapshots of first harmonics of $u_c^{DNS} - u_c^{COX} \rightarrow \hat{u}_c$



Test - evolution in time



Tests - Variation of Bo and a_i



5. High-fidelity data: capillary-driven

Governing Equations:

Two-phase Stokes equations

$$\vec{\nabla} \cdot \vec{u} = 0,$$

$$\rho \frac{\partial \vec{u}}{\partial t} = -\vec{\nabla} p + \vec{\nabla} \cdot \left[\mu \left(\vec{\nabla} \vec{u} + \vec{\nabla} \vec{u}^T \right) \right] + \sigma \kappa \delta_\Gamma \vec{n} + \hat{\rho} \vec{g},$$

$$\frac{\partial C}{\partial t} + \vec{\nabla} \cdot (\vec{u} C) = 0, \text{ where } C(\vec{x}, t) = \begin{cases} 1 & \text{if } \vec{x} \in \text{liquid}, \\ 0 & \text{if } \vec{x} \in \text{gas}. \end{cases}$$

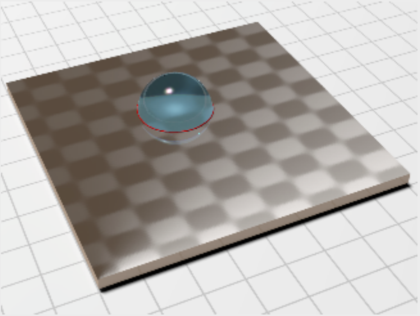
Physical properties ξ calculation: $\xi(\vec{x}, t) = \xi_1 C(\vec{x}, t) + \xi_2 (1 - C(\vec{x}, t)).$

Boundary conditions: impose local contact angle (chemical heterogeneity) on surface.

Data generation - Direct Numerical Simulations

Code: Basilisk

- random heterogeneities, from a 7-parameter functional form
- 10–50 dimensionless times, snapshot saved every 0.1 time units
- adaptive mesh refinement, local grid size between $1/2^5 - 1/2^8$



Dataset:

- 300 DNS cases
- 80,000 contact line snapshots

Reduced-order model

Cox (J. Fluid Mech. 1986)

Analysis near the contact line reveals

$$u_\nu^{COX} = \frac{\sigma}{\mu}\left(\frac{F(\vartheta_*) - F(\theta)}{\ln\left(\frac{\lambda}{r_0}\right) + \frac{Q_o}{f(\theta)} - \frac{Q_i}{f(\vartheta_*)}}\right)$$

- λ, slip length, scales with Δx
- σ surface tension; μ viscosity
- Q_o and Q_i are unspecified
- F and f are known

Analysis near the contact line reveals

$$u_\nu^{COX} = \frac{\sigma}{\mu}\left(\frac{F(\vartheta_*) - F(\theta)}{\ln\left(\frac{\lambda}{r_0}\right) + \frac{Q_o}{f(\theta)} - \frac{Q_i}{f(\vartheta_*)}}\right)$$

- λ, slip length, scales with Δx
- σ surface tension; μ viscosity
- Q_o and Q_i are unspecified
- F and f are known

Obtaining θ assuming quasi-static dynamics

Given $\boldsymbol{c}$, obtain θ from the slope of the solution to the Young–Laplace eqn

$$-\sigma\nabla\cdot\hat{\boldsymbol{n}} = \Delta p, \quad \hat{\boldsymbol{n}} \text{ the surface unit normal}$$

Δp is constant specified by the volume constraint.

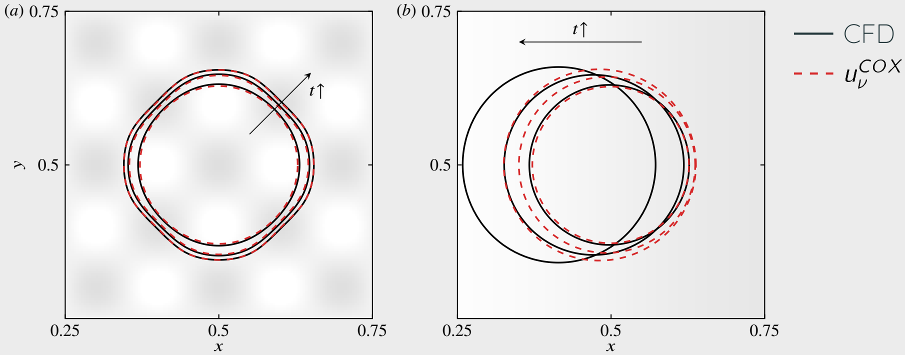
➥ Using the open source code, **Surface Evolver** (SE). Repeated calls to SE during training/testing through a dedicated Python interface.

Cox (J. Fluid Mech. 1986)

Analysis near the contact line reveals

$$u_\nu^{COX} = \frac{\sigma}{\mu}\left(\frac{F(\vartheta_*) - F(\theta)}{\ln\left(\frac{\lambda}{r_0}\right) + \frac{Q_o}{f(\theta)} - \frac{Q_i}{f(\vartheta_*)}}\right)$$

- λ, slip length, scales with Δx
- σ surface tension; μ viscosity
- Q_o and Q_i are unspecified
- F and f are known



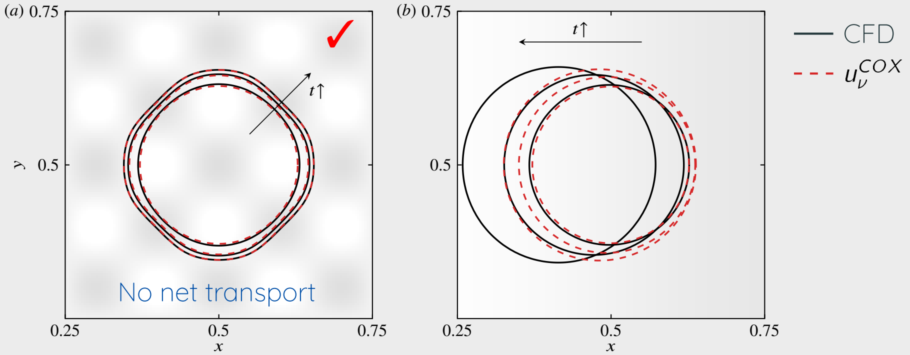
Reduced-order model

Cox (J. Fluid Mech. 1986)

Analysis near the contact line reveals

$$u_\nu^{COX} = \frac{\sigma}{\mu}\left(\frac{F(\vartheta_*) - F(\theta)}{\ln\left(\frac{\lambda}{r_0}\right) + \frac{Q_o}{f(\theta)} - \frac{Q_i}{f(\vartheta_*)}}\right)$$

- λ, slip length, scales with Δx
- σ surface tension; μ viscosity
- Q_o and Q_i are unspecified
- F and f are known



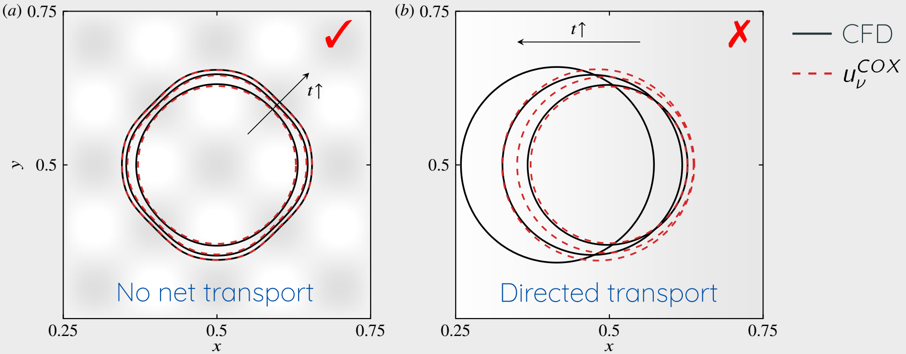
Reduced-order model

Cox (J. Fluid Mech. 1986)

Analysis near the contact line reveals

$$u_\nu^{COX} = \frac{\sigma}{\mu}\left(\frac{F(\vartheta_*) - F(\theta)}{\ln\left(\frac{\lambda}{r_0}\right) + \frac{Q_o}{f(\theta)} - \frac{Q_i}{f(\vartheta_*)}}\right)$$

- λ, slip length, scales with Δx
- σ surface tension; μ viscosity
- Q_o and Q_i are unspecified
- F and f are known



Cox (J. Fluid Mech. 1986)

Analysis near the contact line reveals

$$u_\nu^{COX} = \frac{\sigma}{\mu}\left(\frac{F(\vartheta}{\ln\left(\frac{\lambda}{r_0}\right) -}\right.$$

- λ, slip length, scales with Δx
- osity
- ed

Cox's model underestimates net transport

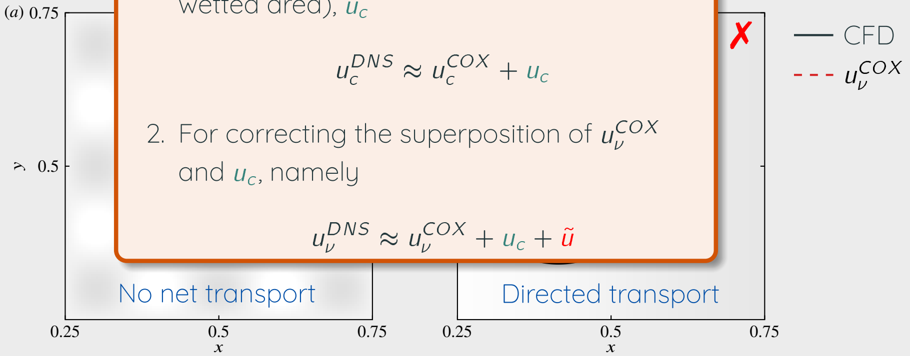
Approach: train two models

1. For net transport (speed of the centroid of wetted area), u_c

$$u_c^{DNS} \approx u_c^{COX} + u_c$$

2. For correcting the superposition of u_ν^{COX} and u_c, namely

$$u_\nu^{DNS} \approx u_\nu^{COX} + u_c + \tilde{u}$$



AI model for correcting net transport motion

Net transport captured by first harmonic; contact line evolves such that contact line has no first harmonic

Input: snapshots of first harmonics of θ and ϑ_*

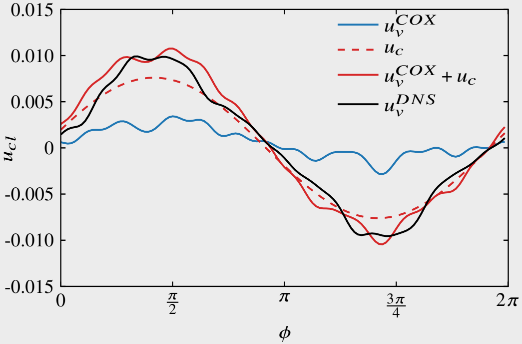
Output: snapshots of first harmonics of $u_\nu^{DNS} - u_\nu^{COX} = u_c^{DNS} - u_c^{COX} \rightarrow u_c$

AI model for correcting net transport motion

Net transport captured by first harmonic; contact line evolves such that contact line has no first harmonic

Input: snapshots of first harmonics of θ and ϑ_*

Output: snapshots of first harmonics of $u_\nu^{DNS} - u_\nu^{COX} = u_c^{DNS} - u_c^{COX} \rightarrow u_c$



AI model for higher-order corrections

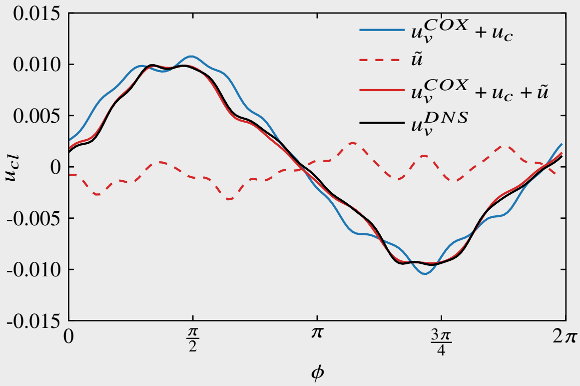
Input: snapshots of $\{c,\ u_\nu^{COX} + u_c\}$

Output: snapshots of $u_\nu^{DNS} - (u_\nu^{COX} + u_c) \rightarrow \tilde{u}$

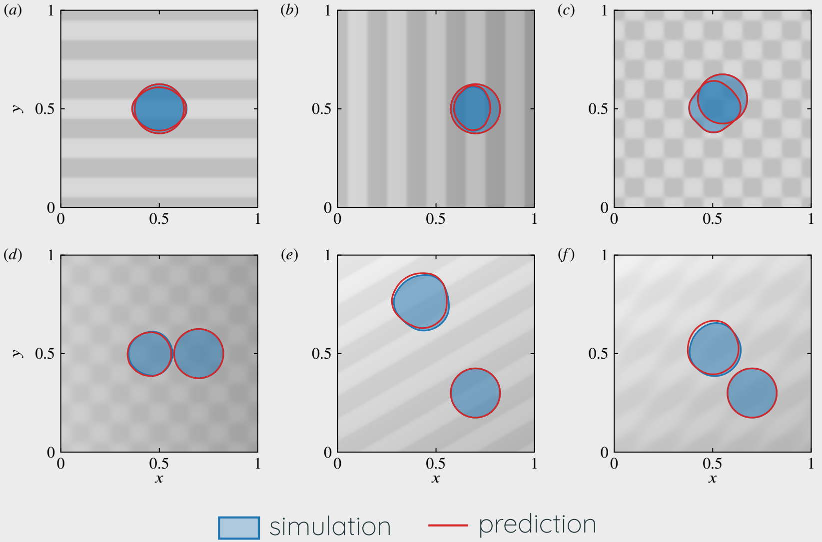
AI model for higher-order corrections

Input: snapshots of $\{c,\ u_\nu^{COX} + u_c\}$

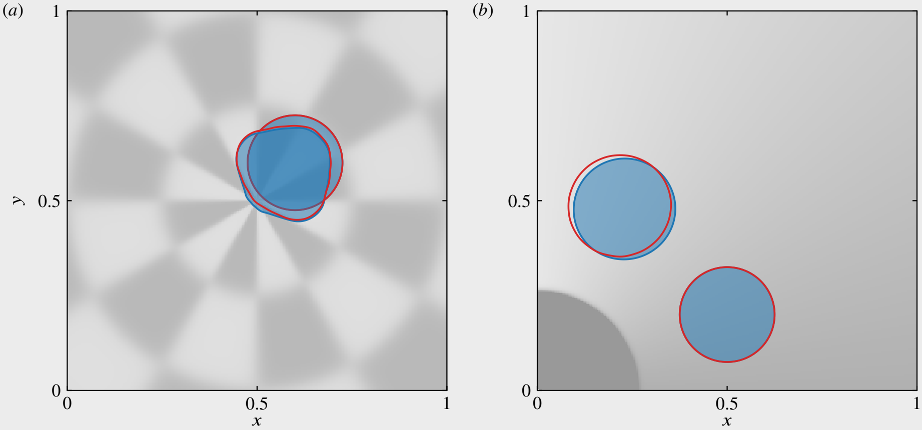
Output: snapshots of $u_\nu^{DNS} - (u_\nu^{COX} + u_c) \rightarrow \tilde{u}$



AI-assisted approach for CFD - Tests



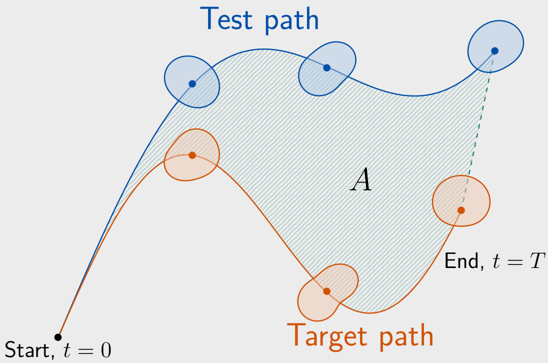
AI-assisted approach for CFD - Out of distribution



Paper in preparation, to be submitted in the coming weeks.

The inverse problem

Given a target droplet path, what heterogenity profile ϑ_* can induce it?



General het. profile given by:

$$\vartheta_* = \sum_{m,n} a_{m,n} \exp(\mathrm{i}k_m x + \mathrm{i}k_n y),$$

$a_{m,n}$ 'design' variables

Optimisation procedure to obtain:

$$a^*_{m,n} = \operatorname{argmin} J$$

where J is a cost function that depends on A and some metric that penalizes non-circular contact lines