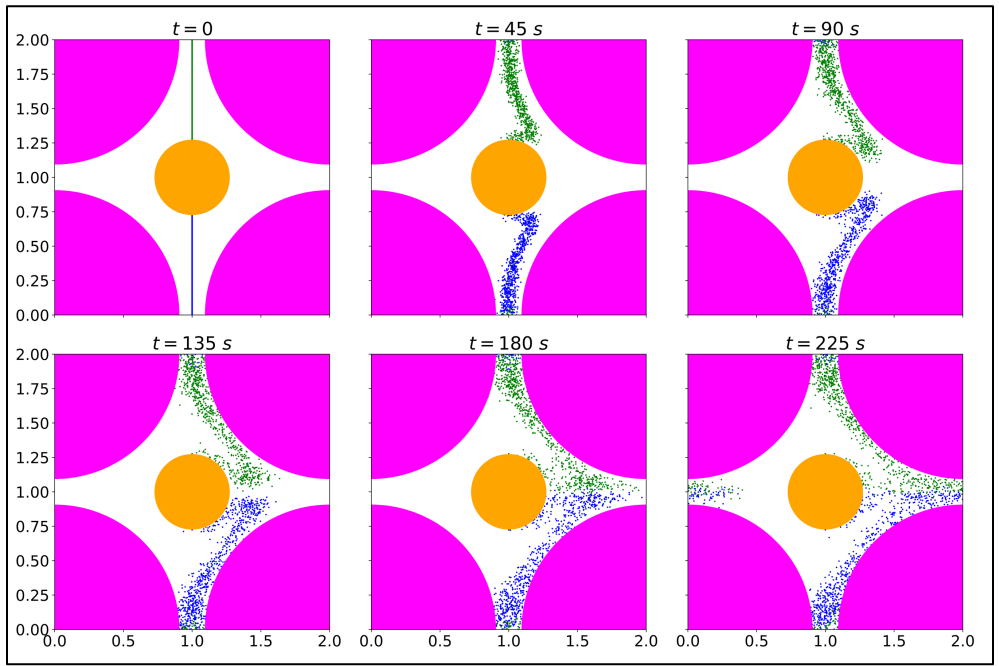
Optimizing Mixing in Porous Media with Automatic Differentiation

Kaylie Hausknecht, Dr. Mohammed Alhashim, Professor Michael Brenner

Harvard University

APS-DFD 2023



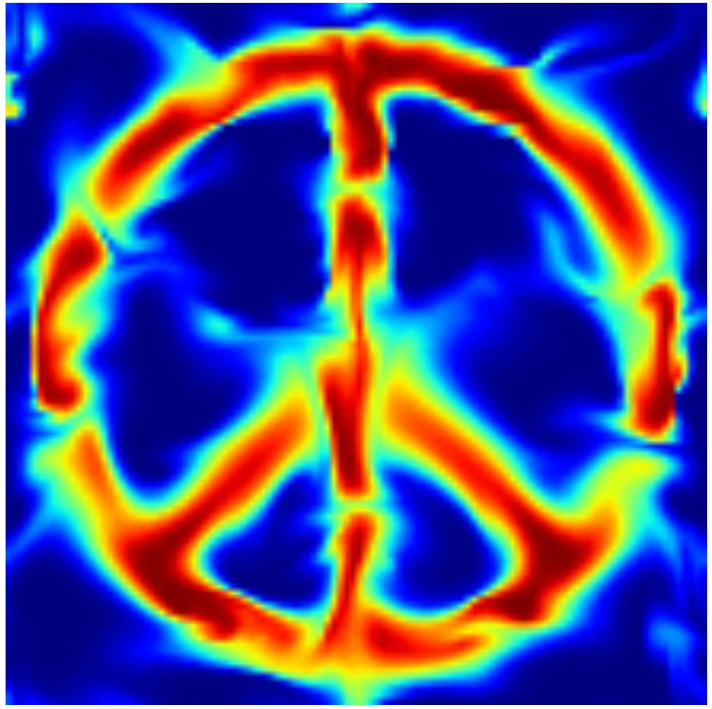
Outline

- Objectives: Inverse Flow Design
- Differentiable Fluid Simulations
  - Taylor Dispersion
- Flow through a porous medium
- Optimization Results
- Conclusions and future work

Goal: Inverse Flow Design

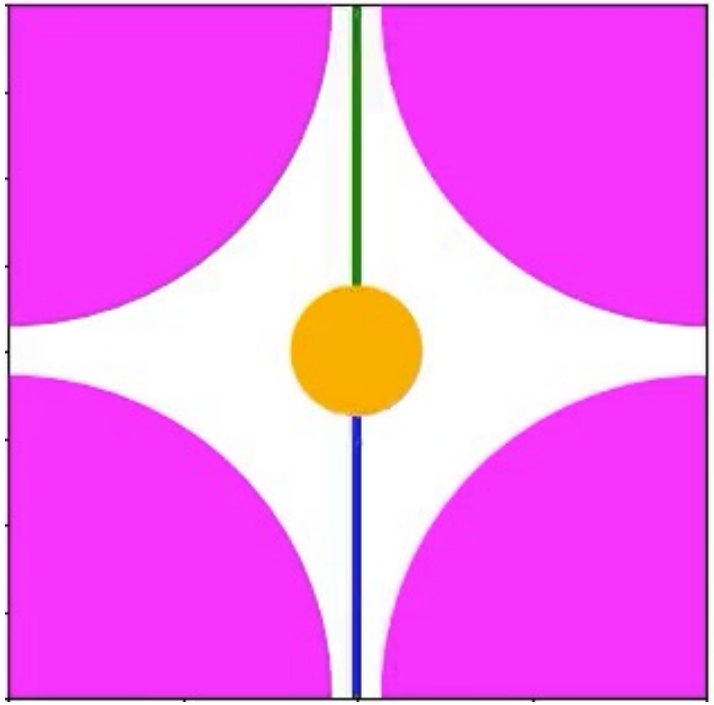
- How can we design fluid setups to target specific flow behavior?

Toy Example



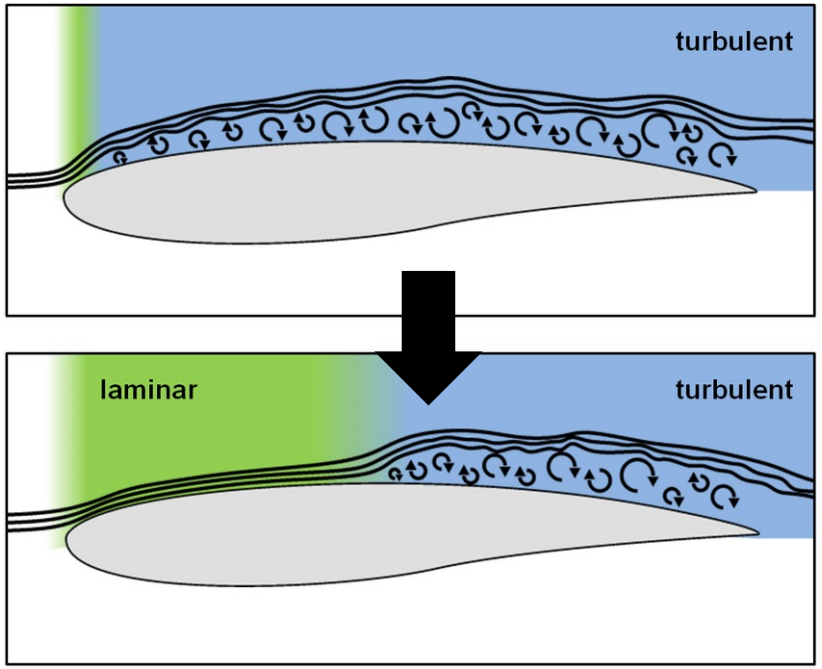
Autograd (2015)

Current Work: Porous Media Design



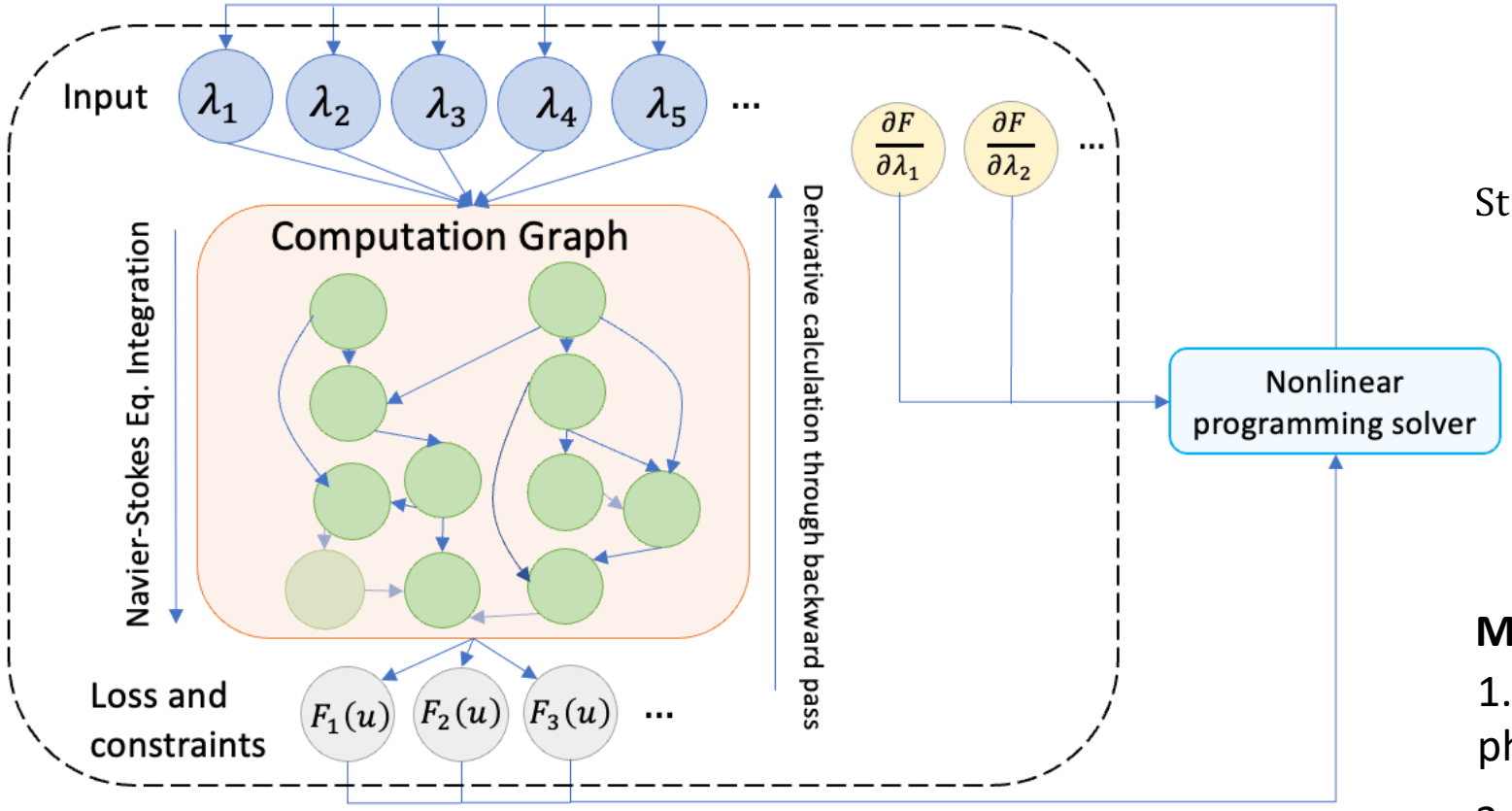
Optimizing Mixing

Potential Future Direction



Approach: Differentiable Physics

Automatic Differentiation



RESEARCH ARTICLE | APPLIED MATHEMATICS

Machine learning–accelerated computational fluid dynamics

Dmitrii Kochkov, Jamie A. Smith, Ayya Alieva, +2, and Stephan Hoyer Authors Info & Affiliations

May 18, 2021 | 118 (21) e2101784118 | https://doi.org/10.1073/pnas.2101784118

Eulerian Problem

Minimize $F(\boldsymbol{u}(t=T))$

Subject to:

$$\mathrm{St}\frac{\partial \boldsymbol{u}}{\partial t} = f(\boldsymbol{u},\boldsymbol{x},t) = -\frac{\partial}{\partial \boldsymbol{x}}\boldsymbol{u}\boldsymbol{u} - \frac{\partial P}{\partial \boldsymbol{x}} + \frac{1}{\mathrm{Re}}\frac{\partial^2 \boldsymbol{u}}{\partial \boldsymbol{x}^2} + \mathbf{f}$$

$$\boldsymbol{u}(t=0) = u_0$$

$$\boldsymbol{u}(\boldsymbol{x}=\Omega) = g(u,x,t)$$

Lagrangian Problem

$$\dot{\boldsymbol{x}} = \boldsymbol{u} + \sqrt{2\boldsymbol{k_B}\boldsymbol{T}\boldsymbol{\xi}(\boldsymbol{t})}$$

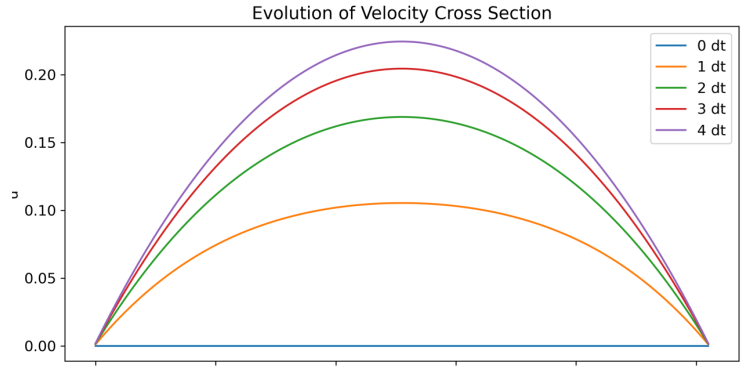
Motivations:

1. Can we introduce noise into differentiable physics frameworks?

2. Develop a versatile toolkit for optimizing over more complex objective functions.

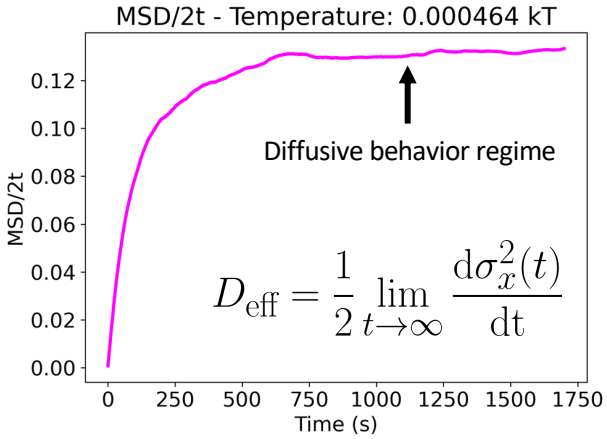
Simulating Taylor Dispersion

1. JAX-CFD - Generate velocity field



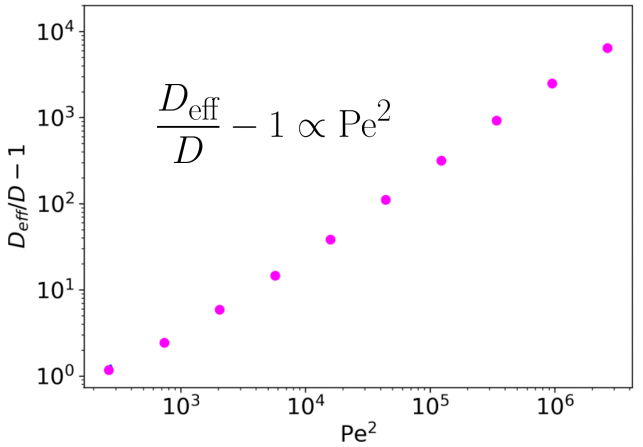
3. Analyze Particle Trajectories

Store particle trajectories, and calculate MSD over time to find



4. Study Scaling with Pe

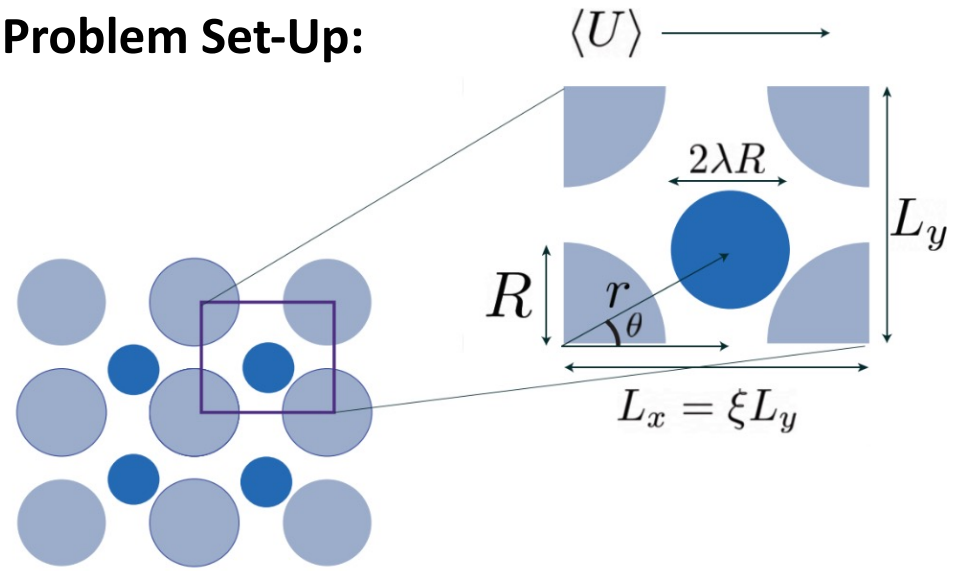
Dispersion coefficient scales according to Taylor Dispersion Theory



2. JAX-MD - Simulate Particle Trajectories

Flow Through a Periodic Porous Medium

Problem Set-Up:

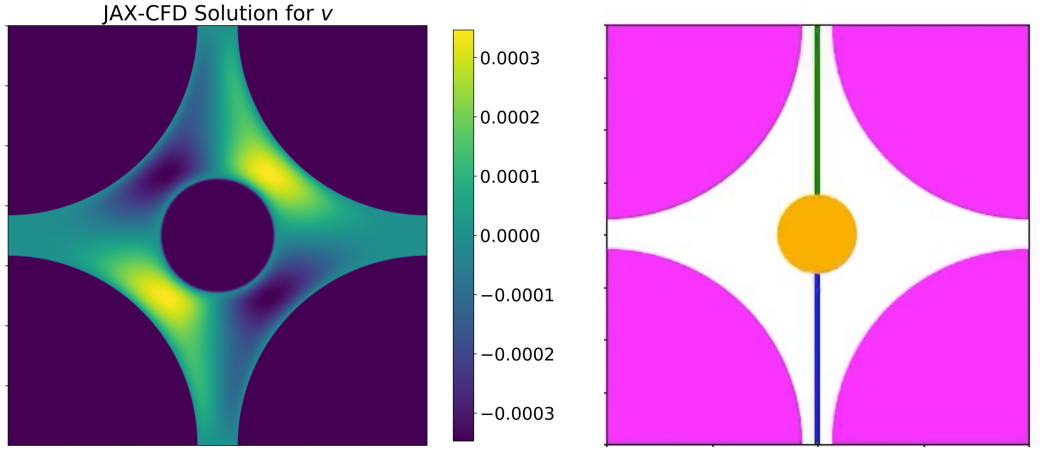


Previous literature has studied dispersion in porous media, but we have never been able to optimize directly over simulations to target a specific dispersion coefficient

Simulations:

① **Differentiable Navier-Stokes Solver**

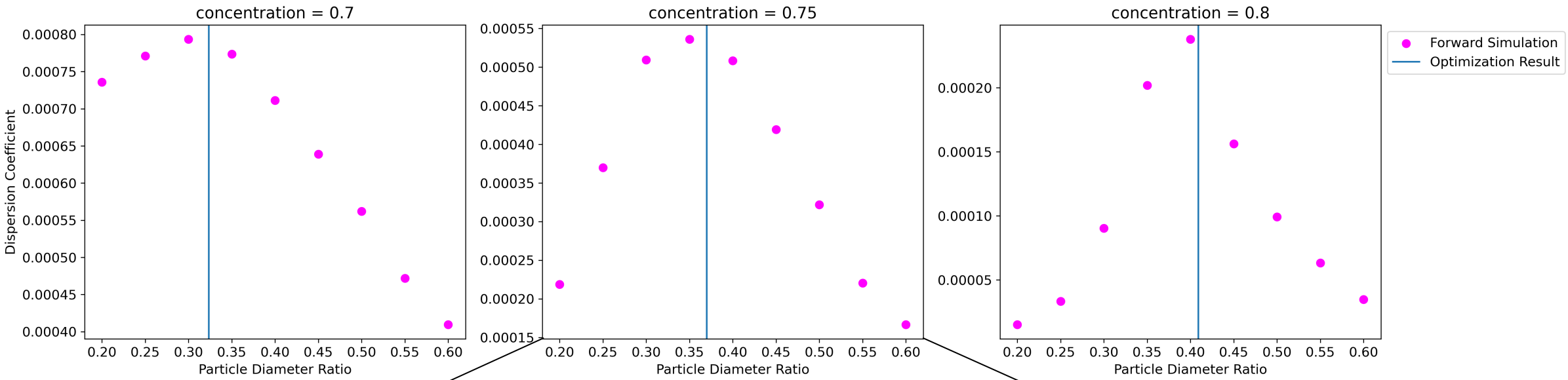
② **Differentiable Brownian Dynamics**



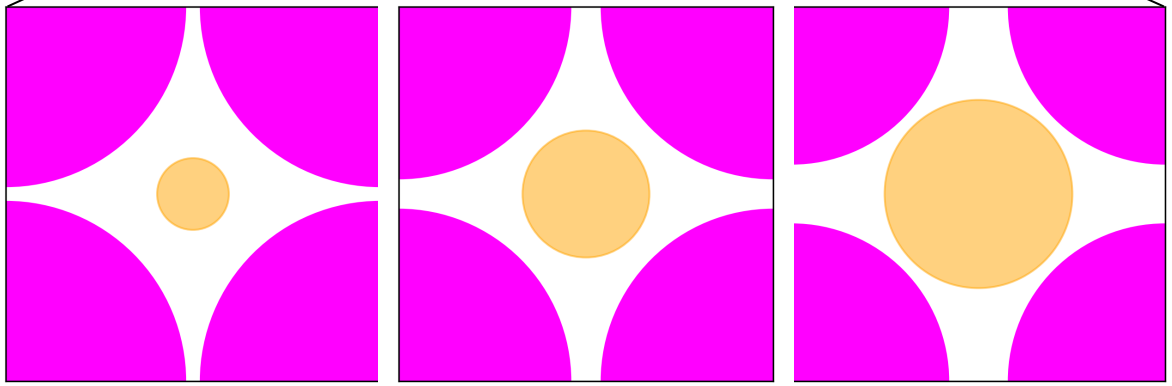
Objective Function:

- Can optimize over arbitrary Lagrangian loss functions
- In this work, we optimize over the dispersion coefficient

Particle Diameter Ratio Optimization



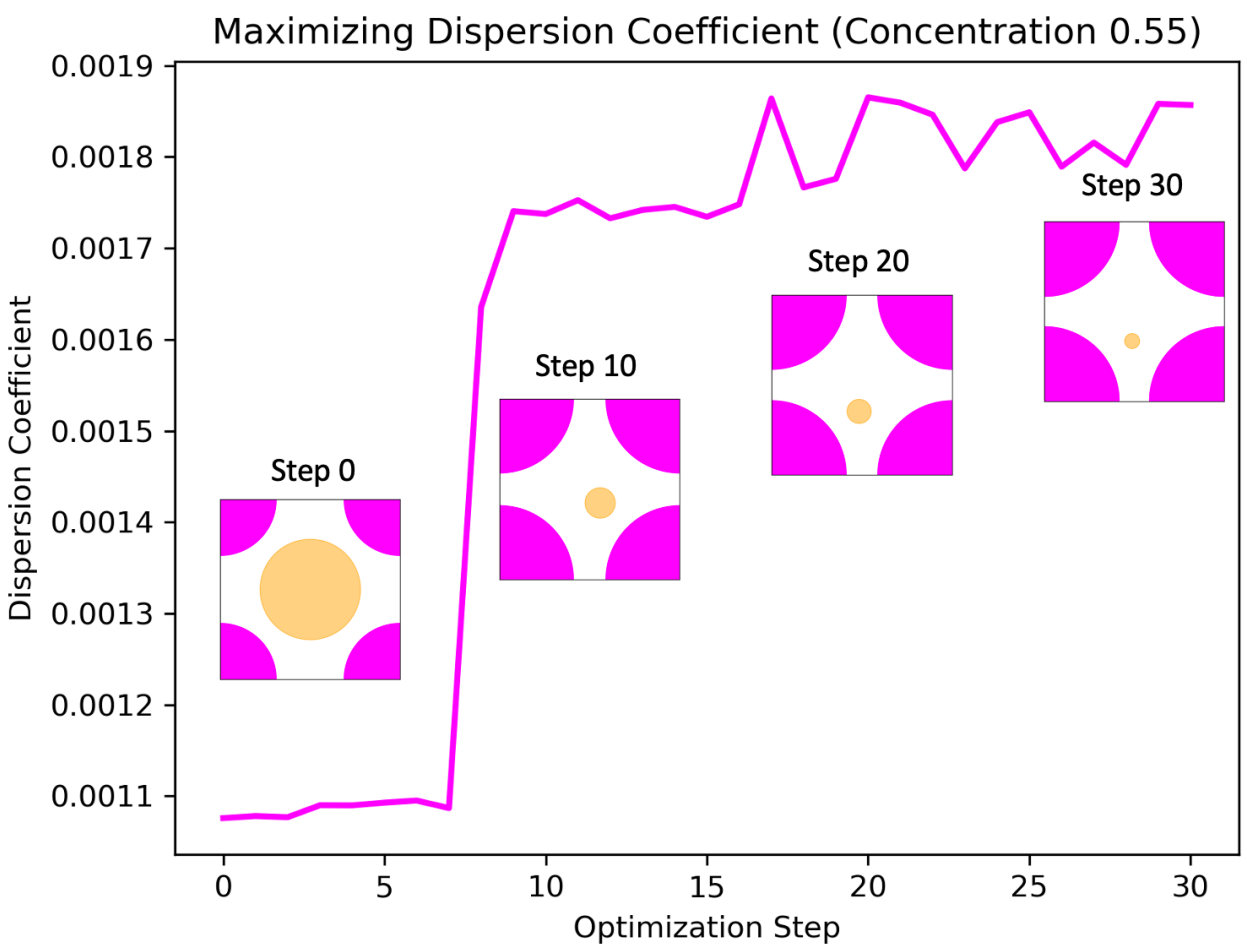
Optimizer matches what we expect from sweeping parameter space for one parameter (particle diameter ratio)

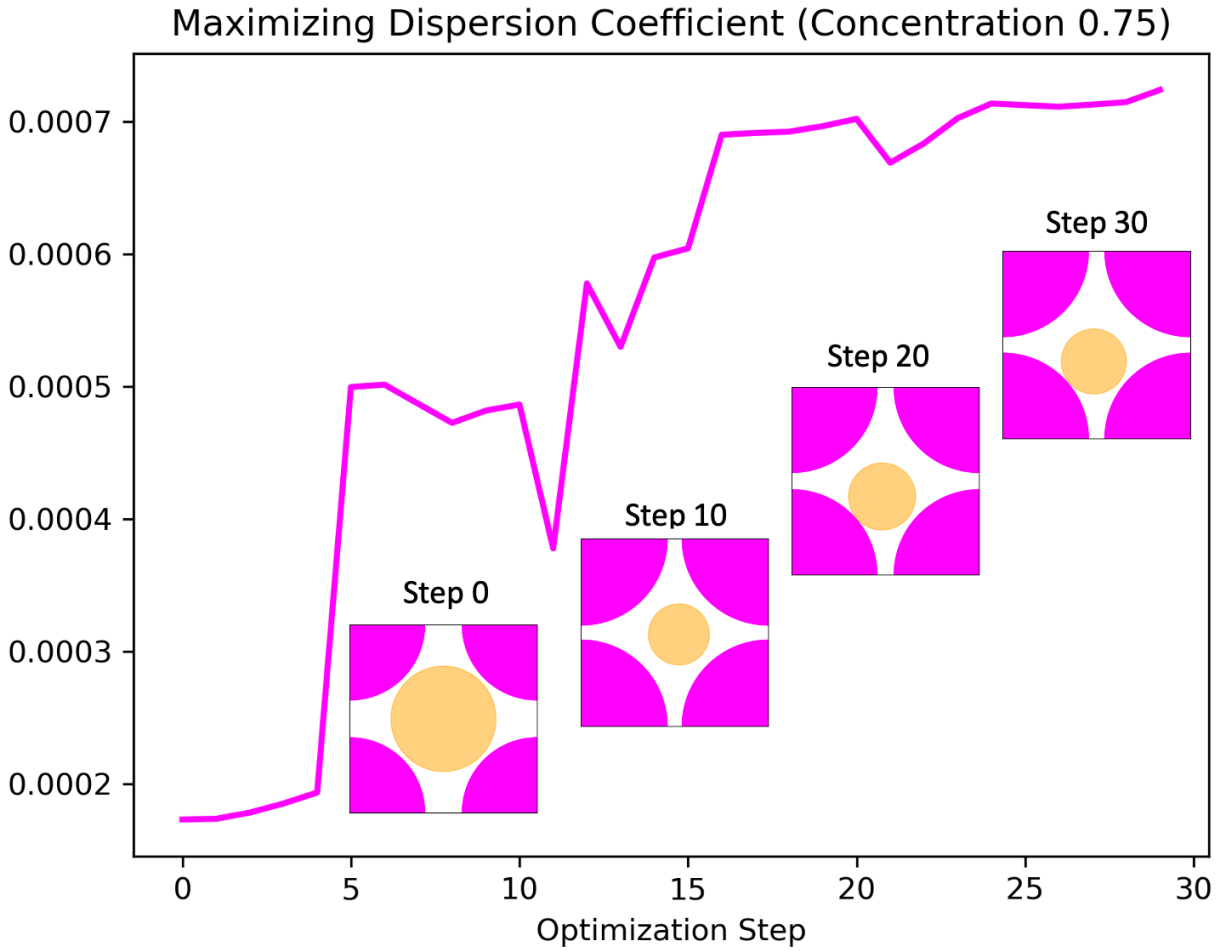


The computational cost of automatic differentiation is independent of the number of design parameters!

Optimizing Dispersion (Pe = 1)

Optimize the dispersion coefficient over **particle diameter ratio**, **x-position of particle**, and **y-position of particle**





Conclusions

- We can combine JAX-CFD and JAX-MD to **study flows from a Lagrangian perspective** and include the **effects of molecular diffusion** in a way that is end-to-end differentiable for steady and unsteady flows
- We can perform novel optimizations of porous media to target specific properties (e.g. mixing)

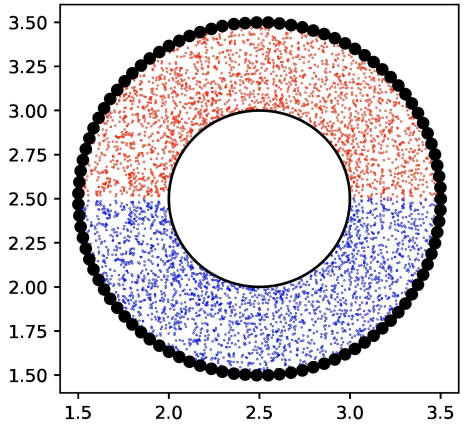
Challenges:

- Calculating dispersion coefficients via Lagrangian simulations is a **multi-scale problem**, and the gradients of the simulation are not well-behaved over very long time scales → limited us to Pe = 1

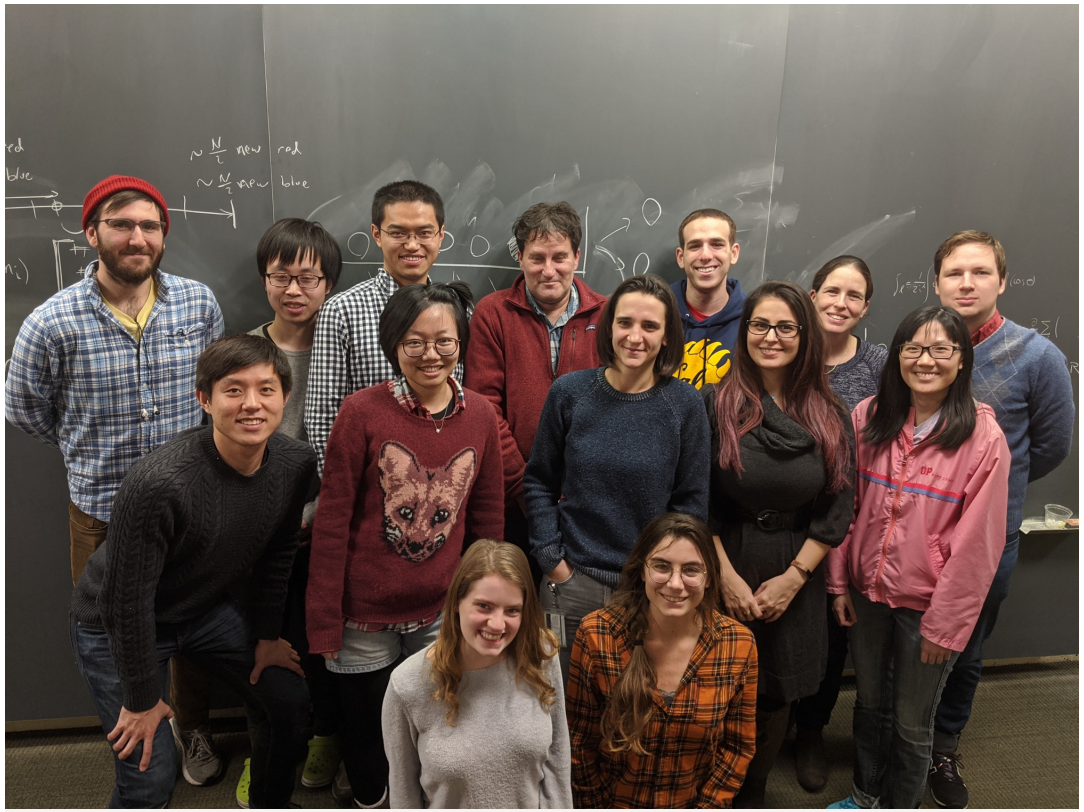
Future Work:

- Study the long-time behavior of automatic differentiation in physics simulations to enable optimization of dispersion in high Péclet number systems
- Apply this methodology to other systems

Optimizing Mixing in Taylor-Couette Flow



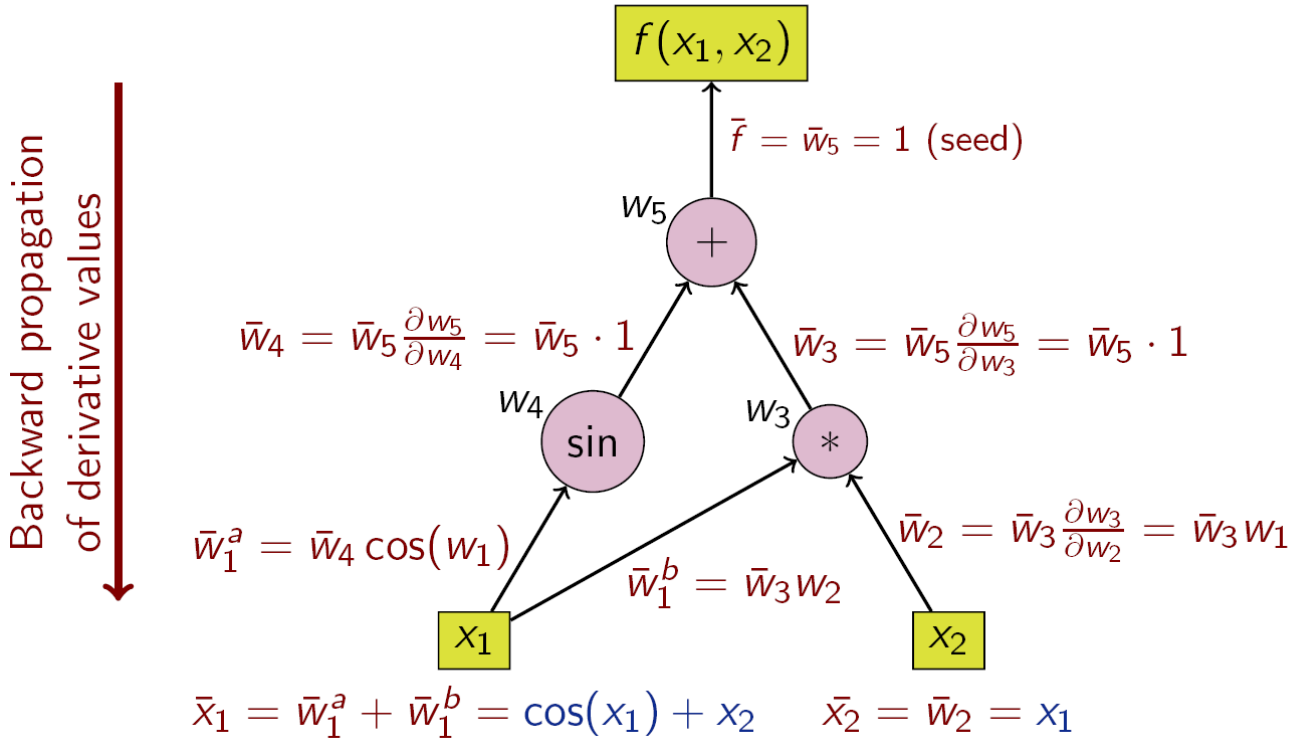
Acknowledgements



Brenner Group
Soft Matter, Biophysics, Applied Math and Materials Science
Harvard School of Engineering and Applied Sciences
Harvard Department of Physics

Appendix 1: Automatic Differentiation

$$y = x_1 x_2 + \sin(x_1)$$
$$= w_1 w_2 + \sin(w_1)$$
$$= w_3 + w_4$$
$$= w_5$$

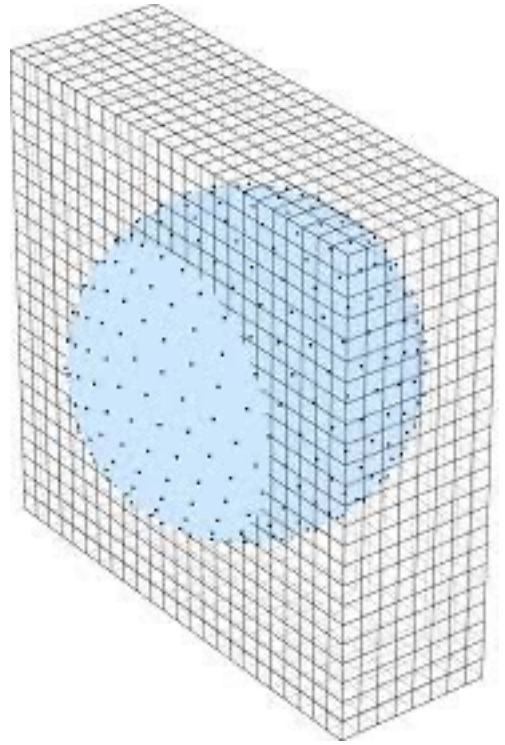


Appendix 2: Immersed Boundary Method

$$\rho\left(\frac{\partial u_i}{\partial t}+\frac{\partial}{\partial x_j}u_i u_j\right)=-\frac{\partial P}{\partial x_i}+\frac{\partial \sigma_{ij}}{\partial x_j}+f_i$$

$$f_i=\int ds\, F_i(s,t)\delta(x_i-X_i)$$

$$\frac{\partial X_i}{\partial t}=U_i=\int dV\, u_i\delta(x_i-X_i)$$



$F_i(s,t)$: any force exerted on particle to resist motion (spring, bending, etc)

Journal of Computational Physics
Volume 209, Issue 2, 1 November 2005, Pages 448-476

An immersed boundary method with direct forcing for the simulation of particulate flows

Markus Uhlmann

Appendix 3: Dispersion Coefficient

$$\mathrm{MSD}(\tau) = \left\langle \left[\vec{r}(t+\tau) - \vec{r}(t)\right]^2 \right\rangle$$
$$= \left\langle \vec{r}(t+\tau)^2 \right\rangle + \left\langle \vec{r}(t)^2 \right\rangle - 2\left\langle \vec{r}(t+\tau)\vec{r}(t) \right\rangle$$

Autocorrelation

Use JAX NumPy FFT algorithm to speed up calculation while maintaining differentiability

*Validated this code by tracking particles with pure diffusion and checking that we derive the diffusion coefficient set for the simulation