Differentiable Fluids with Solid Coupling for Learning and Control

Tetsuya Takahashi (Adobe, University of Maryland at College Park)
Junbang Liang, Yi-Ling Qiao, Ming C. Lin (University of Maryland at College Park)

1

Motivation

2

Previous Work

- Differentiable fluids
  - Adjoint method
  - Neural networks

[McNamara et al. 2004]

[Holl et al. 2020] [Um et al. 2020]

3

Problems

- Lack of solid-to-fluid coupling
- No control method for solids within fluids
- Scalable and efficient differentiable solver

4

Our Approach

- Variational principle for one-way solid-fluid coupling
- Adjoint method applied to the entire simulation steps
- Neural networks for control force learning

5

Method

6

Our Method

- One-way fluid-solid coupling
- Adjoint method for gradient computation
- Neural networks for control force learning

7

One-way fluid-solid coupling

- Divergence-free constraints
- Free-slip boundary condition
- Variational principle: optimization for pressure

Objective function for fluids Solid velocity

$$p = \text{argmin}\ (E_f(\boldsymbol{u}^t, p) + E_r(\boldsymbol{v}^t, p))$$

Pressure Fluid velocity Objective function for solids

8

Incompressible Velocity Fields

- Optimization for pressure

$$p = \text{argmin}\,(E_f(\boldsymbol{u}^t, p) + E_r(\boldsymbol{v}^t, p))$$

- Correct fluid velocity

$$\boldsymbol{u}^{t+\Delta t} = \boldsymbol{u}^t - \Delta \boldsymbol{P}^{-1}\boldsymbol{G}p$$

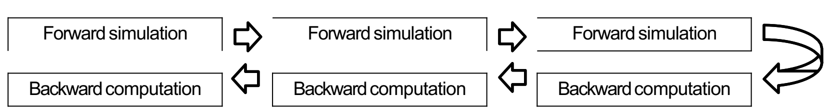
- Matrix form notation

$$\begin{pmatrix} \boldsymbol{u}^{t+\Delta t} \\ \boldsymbol{v}^{t+\Delta t} \end{pmatrix} = \begin{pmatrix} \boldsymbol{B} & \boldsymbol{C} \\ \boldsymbol{O} & \boldsymbol{I} \end{pmatrix} \begin{pmatrix} \boldsymbol{u}^t \\ \boldsymbol{v}^t \end{pmatrix}$$

10

Gradient Computation with Adjoint Method

- Costly gradient computation
  - Expensive finite-difference approximation
  - High overhead of low-level automatic differentiation
- Adjoint method
  - Forward state update
  - Backward adjoint state update



10

Backward Pressure Solve

- Forward pressure solve with velocity update

$$\begin{pmatrix} \boldsymbol{u}^{t+\Delta t} \\ \boldsymbol{v}^{t+\Delta t} \end{pmatrix} = \begin{pmatrix} \boldsymbol{B} & \boldsymbol{C} \\ \boldsymbol{O} & \boldsymbol{I} \end{pmatrix} \begin{pmatrix} \boldsymbol{u}^{t} \\ \boldsymbol{v}^{t} \end{pmatrix}$$

- Backward adjoint state update

⇓ Transpose

$$\begin{pmatrix} \hat{\boldsymbol{u}}^{t} \\ \hat{\boldsymbol{v}}^{t} \end{pmatrix} = \begin{pmatrix} \boldsymbol{B} & \boldsymbol{O} \\ \boldsymbol{C} & \boldsymbol{I} \end{pmatrix} \begin{pmatrix} \hat{\boldsymbol{u}}^{t+\Delta t} \\ \hat{\boldsymbol{v}}^{t+\Delta t} \end{pmatrix}$$

11

Neural Net Integration

- Control input predictions
- Back propagation through neural networks

12

Result

13

Implementation and Experiment Details

- C++ implementation
- Pytorch 1.5 for neural networks
- Intel Core i5-7200U with 8GB RAM

14

Comparison to Numerical Differentiation

15

Comparison to PhiFlow

Our method is 1-2 orders of magnitude faster by avoiding low-level automatic differentiation employed in PhiFlow.

16

Comparison to Low-Level Automatic Differentiation

Our method is 1-2 orders of magnitude faster by avoiding low-level automatic differentiation.

Differentiable Fluids with Solid Coupling for Learning and Control

Tetsuya Takahashi[*,†], Junbang Liang[†], Yi-Ling Qiao[†], Ming C. Lin[†]

[*]Adobe, [†]University of Maryland at College Park

Summary

- Variational formulation for one-way solid-fluid coupling
- Adjoint method for gradient computation
- Learning and control framework with neural networks

22

Future Work

- Extensions to viscous fluids and deformable solids
- Experiments in the real-world environments

23

Acknowledgment

- National Science Foundation
- Elizabeth Stevinson Iribe Chair Professorship
- Anonymous reviewers

24