Paper A: A Dimension-reduced Pressure Solver for Liquid Simulations

Ryoichi Ando, Nils Thürey, and Chris Wojtan
IST Austria & TMU

Reporter: Xiangyun Xiao

Date: 2018-10-24

Abstract简析(贡献)：

- This work presents a method for efficiently simplifying the pressure projection step in a liquid simulation. We first devise a straightforward **dimension reduction technique that dramatically reduces the cost of solving the pressure projection**. Next, we **introduce a novel change of basis that satisfies free-surface boundary conditions exactly, regardless of the accuracy of the pressure solve.** When combined, these ideas greatly reduce the computational complexity of the pressure solve without compromising free surface boundary conditions at the highest level of detail. Our techniques are easy to parallelize, and they effectively eliminate the computational bottleneck for large liquid simulations.
- 首先，文章设计一种比较直观的降维方法(dimension reduction) ----- 可以有效的减少projection的solving cost;
- 其次，在不考虑pressure solve 的accuracy 情况下（而例如，2010年SIGGRAPH 上也有一篇论文类似于本文，但需要额外的高精度的pressure solve 去enforce free-surface boundary），提出a novel change of basis 去满足free-surface boundary conditions;

Related work(几个重要的相关工作):

- [LZF10]: A novel algorithm for incompressible flow using only a coarse grid projection ----- 2010年Ron组的，coarse grid projection, 但是需要对free-surface 高精度处理；
- Multi-grid: 效果很好，特别是在求解Poisson problems. 但是“in practice its convergence strongly hinges on an accurate discretization of the problem on the coarser grids in the hierarchy”；
- Model reduction: [TLP06],[WST09], etc.
- Add detail to low-res sim: [KTJ08], etc.
- [EB14]: Detailed water with coarse grids: Combining surface meshes and adaptive discontinuous galerkin; 2014年Bridson组的，use higher-order polynomial bases for liquid simulation; 本文用fast tri-linear interpolation；
- [NB11]: Guide shapes for high resolution naturalistic liquid simulation; 用低精度sim去指导高精度sim;可以作为本文的互补；

A dimension-reduced pressure solver:

- N-S equation: $$\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u} \cdot \nabla \boldsymbol{u} = -\frac{1}{\rho} \nabla p + \boldsymbol{f}, \qquad \nabla \cdot \boldsymbol{u} = 0$$
- Pressure field : $$\operatorname*{argmin}_{p} \int_{\text{fluid}} \frac{\rho}{2} \left\| \breve{\boldsymbol{u}} - \frac{\Delta t}{\rho} \nabla p \right\|^2 dV$$

$$\frac{\Delta t}{\rho} [\nabla^2] p = [\nabla]^T \breve{\boldsymbol{u}} \, .$$

Poisson equation

Here $\breve{\boldsymbol{u}}$ is the intermediate velocity after the advection, Δt is the time step size, $[\nabla]$ is the discretized gradient operator, and $[\nabla^2] \equiv [\nabla]^T [\nabla]$ is the discretized Laplacian operator. We use the level set method to track the liquid free surface [OF03], and we enforce the Dirichlet boundary conditions with the ghost fluid method [GFCK02].

A dimension-reduced pressure solver:

- To reduce the degrees of freedom in the pressure projection:

$$p = U\tilde{p}$$

a) $\tilde{p}$ -- a pressure field sampled on a coarse grid;

b) U -- sparse up-sampling matrix that interpolates $\tilde{p}$ onto hi-res grid;

c) Use linear interpolation(其他高维插值也行，U会更密，conditioning poor);

- Instead of applying $p = U\tilde{p}$ to $\frac{\Delta t}{\rho}[\nabla^2]p = [\nabla]^T \check{\boldsymbol{u}}$. (导致超定方程·多解).
- Substitute $p = U\tilde{p}$ into $\underset{p}{\operatorname{argmin}} \int_{\text{fluid}} \frac{\rho}{2} \left\| \check{\boldsymbol{u}} - \frac{\Delta t}{\rho} \nabla p \right\|^2 dV$, 得到:

$$U^T \frac{\Delta t}{\rho}[\nabla^2]U\tilde{p} = U^T[\nabla]^T \check{\boldsymbol{u}}.$$

Least square system

- Compute $\tilde{p}$ using MIC(0)-PCG [Bri08]:

$$[\nabla]p = [\nabla]U\tilde{p} \qquad \frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u} \cdot \nabla \boldsymbol{u} = -\frac{1}{\rho}\nabla p + \boldsymbol{f}, \qquad \nabla \cdot \boldsymbol{u} = 0$$

A dimension-reduced pressure solver:

- naive down-sampling method:

  a) generate a low-resolution velocity with $\mathbf{u}_L = U^T \breve{\boldsymbol{u}}$;

  b) Solving U $\frac{\Delta t}{\rho}[\nabla^2]p = [\nabla]^T \breve{\boldsymbol{u}}$. on a coarse grid.

  c) Up-sample a corrected velocity with $U\mathbf{u}_L$.

- Poor performance:

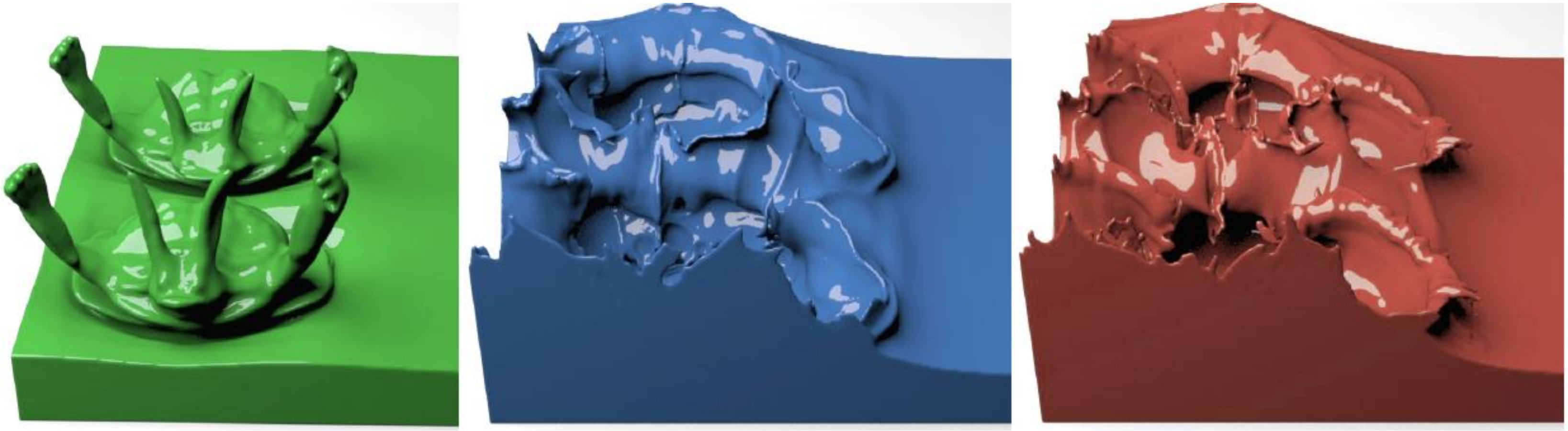


Figure 2: *From left to right: a naively down-sampled coarse pressure solver; our new dimension-reduced pressure solver coupled with higher resolution levelset; a full resolution simulation. A resolution of $257 \times 205 \times 257$ was used for velocity and levelset in all cases, while the coarsened versions use a $17 \times 13 \times 17$ grid for pressure (reduced by a factor of 16^3).*

A dimension-reduced pressure solver:

- The reduced system:

$$U^T \frac{\Delta t}{\rho}[\nabla^2]U\tilde{p} = U^T[\nabla]^T \check{\boldsymbol{u}}.$$

- A **denser matrix** that consists of interpolated versions of all the original equations;
- It computes a solution that simultaneously accommodates the high-resolution boundary conditions as accurately as possible(**在最小二乘意义下，而非直接采样计算**);

- However:
  - Solving $\tilde{p}$ from the reduced system, 相比于精细的liquid surface, degree of freedom 会少的多，导致对于平滑边界处理效果还好，但如果对于复杂边界，会出现问题（维度不对等）（these mismatched degrees of freedom can slightly violate Dirichlet conditions

A novel surface-aware pressure basis

- p = 0 at the liquid surface, force our reduced pressure to meet this condition by directly leveraging our high-resolution surface geometry.
- Encode the pressure as a scalar multiple of the signed distance function:

$$p = \Phi U \tilde{p}$$

$$p = \hat{\Phi} U \tilde{p} \qquad \hat{\phi} = \sin\left(\frac{\pi}{2}\min\left(1, \frac{-\phi}{2\Delta x}\right)\right)$$

$$U^T \hat{\Phi} \frac{\Delta t}{\rho} [\nabla^2] \hat{\Phi} U \tilde{p} = U^T \hat{\Phi} [\nabla]^T \check{\boldsymbol{u}}.$$

Modified Poisson equation

Surface tension：

- For liquids with nonzero surface tension, the Dirichlet condition changes to:

$$p = \sigma H$$

σ 表面张力，H 平均曲率；

- 加上表面张力项，p 就为：

$$p = \hat{\Phi} U \tilde{p} + (I - \hat{\Phi}) \sigma H$$

在free-surface上则 $p = \sigma H$，而在流体内部则：$p = U\tilde{p}$

- 然后，求解：

$$\underset{\tilde{p}}{\operatorname{argmin}} \int_{\text{fluid}} \frac{\rho}{2} \left\| \left(\check{\boldsymbol{u}} - \frac{\Delta t}{\rho} [\nabla](I - \hat{\Phi}) \sigma H \right) - \frac{\Delta t}{\rho} [\nabla] \hat{\Phi} U \tilde{p} \right\|^2 dV$$

- Finds the unique pressure $\tilde{p}$ which minimizes the kinetic energy of the

3 Jacobi iterations：

- $\tilde{p}$ 其实是不满足divergence –free constraint in hi-res.
- 文章用了post-processing ---- 3 Jacobi iterations on:

$$\frac{\Delta t}{\rho}[\nabla^2]p = [\nabla]^T \check{\boldsymbol{u}} \, .$$

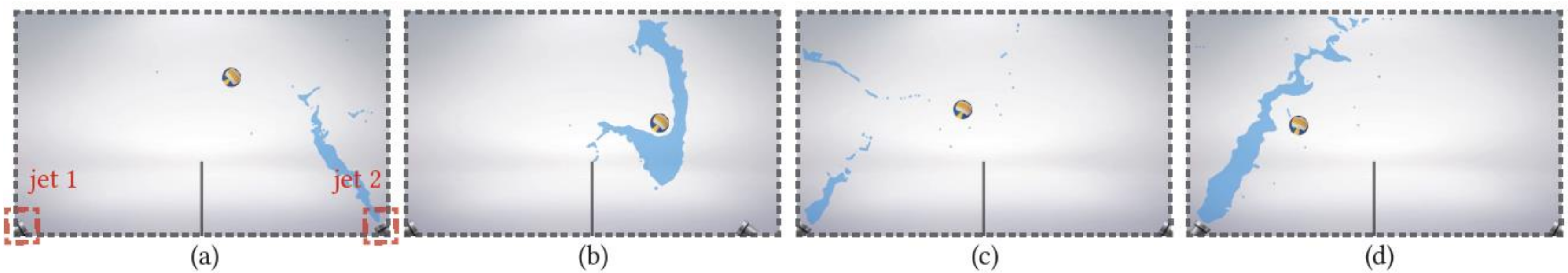
- Remove high-frequency artifacts (divergence);
- Qualitatively indistinguishable from a full high-resolution solution;

Paper B: Fluid Directed Rigid Body Control using Deep Reinforcement Learning (Autoencoder)

Pingchuan Ma, Yunsheng Tian, Zherong Pan, Bo Ren, Dinesh Manocha
Nankai University & UNC & UMCP

工作简析(贡献)：

- 本文的主要目标（举例）：



- 学习一个控制系统，去控制a coupled **2D system** involving both fluid and rigid bodies. 如上图示例。
- 系统包括两个部分：Training stage & generation stage;
- 主要关注Training stage: two-block neural-net:
  - **a) extract the low-dimensional features from high-dim fluid states;**
  - **b) map the features to control inputs**

  - 其中，我们关注，**auto-encoder去做features extraction**

Auto-encoder:

- System equations：

$$\forall \mathbf{x} \in \Omega_f : \begin{cases} \dot{\mathbf{u}} + (\mathbf{u} \cdot \nabla)\mathbf{u} = -\nabla p + \mathbf{f} \quad \nabla \cdot \mathbf{u} = 0 \\ \dot{\rho} + (\mathbf{u} \cdot \nabla)\rho = 0 \end{cases}$$

N-S equation

$$\forall \mathbf{x} \in \Omega_r : \begin{cases} \mathbf{u} = \mathbf{v} + \omega \times (\mathbf{x} - \mathbf{o}) \\ \mathbf{M}\begin{pmatrix} \dot{\mathbf{v}} \\ \dot{\omega} \end{pmatrix} + \begin{pmatrix} \int_{\partial\Omega_r} \mathbf{p} ds \\ \omega \times \mathbf{I}\omega + \int_{\partial\Omega_r} \mathbf{p}(\mathbf{x} - \mathbf{c}) \times ds \end{pmatrix} = 0 \\ \begin{pmatrix} \dot{\mathbf{c}} \\ \dot{\mathbf{R}} \end{pmatrix} = \begin{pmatrix} \mathbf{v} \\ [\omega]\mathbf{R} \end{pmatrix} \end{cases}$$

Newton-Euler's equation

$$\forall \mathbf{x} \in \Omega_f \cup \Omega_r : \quad \mathbf{u} \in C^0.$$

- Independent variables: **u**, ρ, **f**, **v**, $\boldsymbol{\omega}$, **c**, **R**
- Dependent variable: **p**
- Denote f as a control variable (可以直接被赋值 modified).
- Express f as a function **f(I)**, **I** representing example-dependent information.

Auto-encoder:

- The final state vector：

$$\mathbf{S} = (\mathbf{u}, \rho, \mathbf{v}, \ \boldsymbol{\omega}, \mathbf{c}, \mathbf{R}, \mathbf{I})$$

- Represent the coupled fluid/rigid simulator as:

$$\left(\mathbf{u}\ \rho\ \mathbf{v}\ \omega\ \mathbf{c}\ \mathbf{R}\right)_{i+1} = \mathcal{F}_{f,r}(\left(\mathbf{u}\ \rho\ \mathbf{v}\ \omega\ \mathbf{c}\ \mathbf{R}\right)_i, \mathbf{f}(\mathbf{I}_i)) \quad \mathbf{I}_{i+1} = \mathcal{F}_{\mathbf{I}}(\mathbf{I}_i, \mathbf{a}_i),$$

- Neural-net Controller:

$$\pi(\mathrm{a}|\mathrm{S}, \theta)$$

  Maps from **S** (state vector) to the low-dim action **a;**
- In previous works [Peng et al. 2017; Won et al. 2017], S is low-dim, π is parametrized using MLP;
- 如果照搬，则会：
  - Too many parameters $\boldsymbol{\theta}$**;**
  - Require a large amount of data;

Auto-encoder:

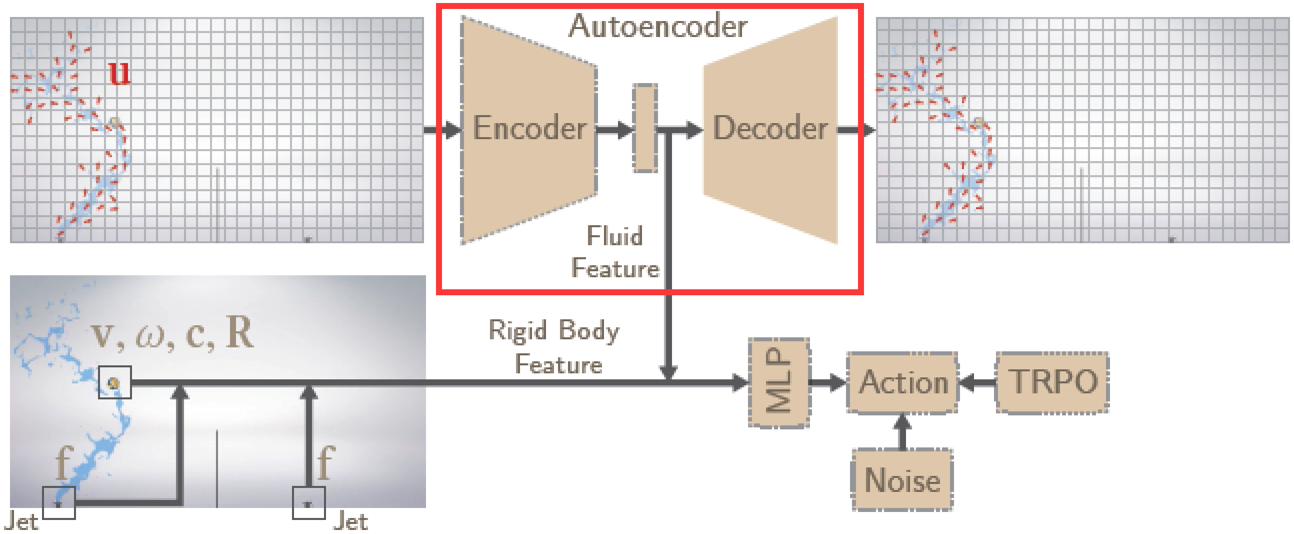
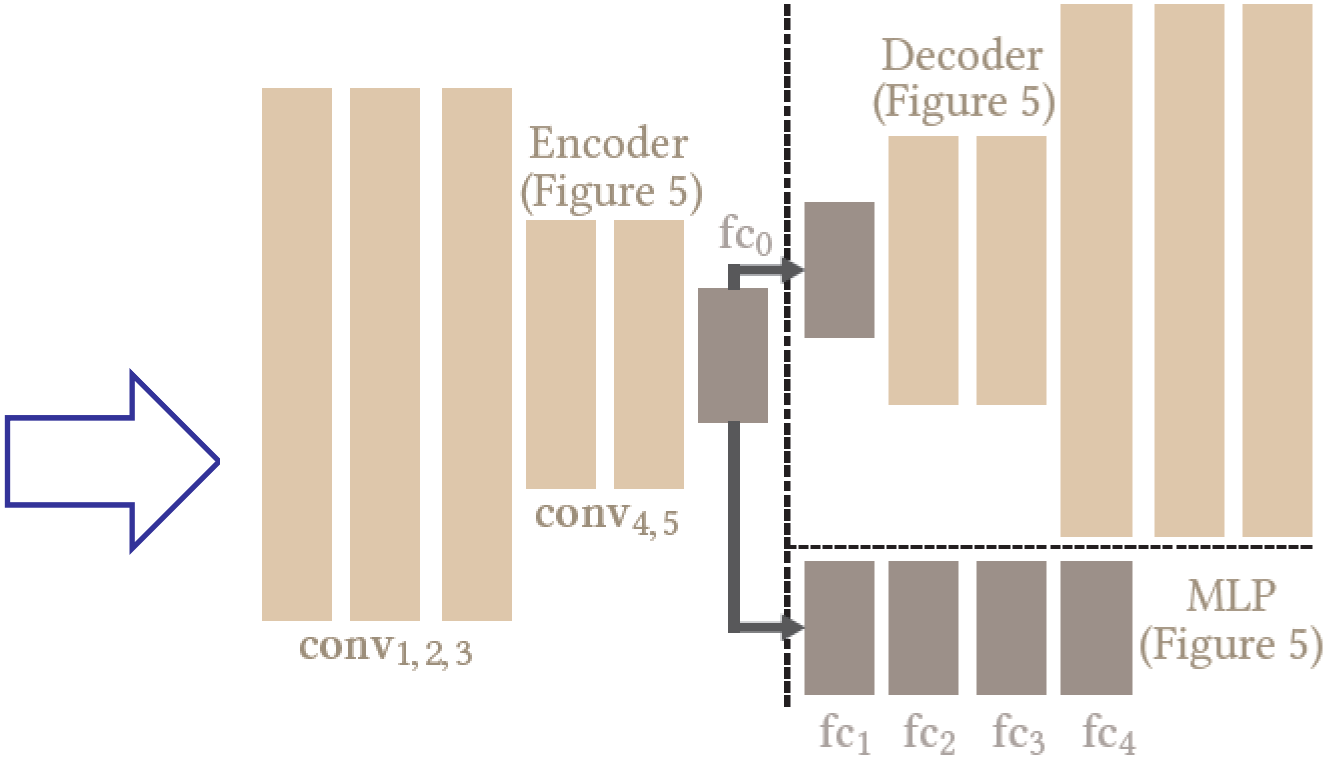


Fig. 5. We illustrate the state variables u, v, ω, c, and R. We illustrate the external force f, which is non-zero only in the source regions (the water jet in the black box). We first extract the fluid's velocity field using a convolutional autoencoder (Section 4.1). The encoded features and other rigid features are combined and then feed into the multilayer perceptron (MLP) to get the action. The MLP and the encoder are trained using a DRL algorithm [Schulman et al. 2015a] (Section 4.2), which is marked with dashed lines.

Auto-encoder:



Encoder : $\mathcal{E}(\bullet, \theta_{\mathcal{E}}),$

Decoder : $\mathcal{D}(\bullet, \theta_{\mathcal{D}}),$

Loss : $\underset{\theta_{\mathcal{E}}, \theta_{\mathcal{D}}}{\operatorname{argmin}} \sum_{i=1}^{N} \|\mathcal{D}(\mathcal{E}(\mathrm{u}_i, \theta_{\mathcal{E}}), \theta_{\mathcal{D}}) - \mathrm{u}_i\|^2.$

Reduced state feature : $\bar{\mathrm{S}} = \left(\mathcal{E}(\mathrm{u})\ \mathrm{v}\ \omega\ \mathrm{c}\ \mathrm{R}\ \mathrm{I}\right) \quad |\bar{\mathrm{S}}| < 100$

Thanks

Q&A