DiffFR: Differentiable SPH-based Fluid-Rigid Coupling for Rigid Body Control - Supplementary

ZHEHAO LI, University of Science and Technology of China, China
QINGYU XU, University of Science and Technology of China, China
XIAOHAN YE, TMCC, College of Computer Science, Nankai University, China
BO REN*, TMCC, College of Computer Science, Nankai University, China
LIGANG LIU, University of Science and Technology of China, China

ACM Reference Format:
Zhehao Li, Qingyu Xu, Xiaohan Ye, Bo Ren, and Ligang Liu. 2023. DiffFR: Differentiable SPH-based Fluid-Rigid Coupling for Rigid Body Control - Supplementary. *ACM Trans. Graph.* 42, 6, Article 179 (December 2023), 6 pages. https://doi.org/10.1145/3618318

In this supplementary, we provide more technical details of the gradient computation of our differentiable SPH-based fluid-rigid coupling simulator and experiment settings. We hope this supplementary material to be self-contained and helpful for implementation.

1 COMPUTATION DETAILS OF GRADIENTS OF RIGID BODY DYNAMICS

For rigid body dynamics, we adopt the semi-implicit integration:

$$v^{n+1} = v^n + \Delta t \mathrm{M}^{-1} f^n, \tag{1}$$

$$x^{n+1} = x^n + \Delta t v^{n+1}, \tag{2}$$

$$\omega^{n+1} = \omega^n + \Delta t (\mathrm{I}^n)^{-1}(L^n \times \omega^n + \tau^n), \tag{3}$$

$$q^{n+1} = \mathrm{normalize}(q^n + \frac{\Delta t}{2}([0, \omega^{n+1}] \otimes q^n)), \tag{4}$$

where Δt is the time step, M a positive diagonal mass matrix, f and τ the external force and torque, I and $L = \mathrm{I} \cdot \omega$ the inertia tensor and angular momentum of the rigid body, respectively. In forward-mode differentiation, given $\frac{\mathrm{d}x^n}{\mathrm{d}s^0}$, $\frac{\mathrm{d}v^n}{\mathrm{d}s^0}$, $\frac{\mathrm{d}q^n}{\mathrm{d}s^0}$, $\frac{\mathrm{d}\omega^n}{\mathrm{d}s^0}$ from the last time step, we need to compute the gradient of rigid body state $\mathbf{s}_{\mathcal{R}}^{n+1}$ at time step $n+1$ with respect to the initial state variable $\mathbf{s}_{\mathcal{R}}^0$ at time step 0 in a simulation trajectory, where $s^0 \in \mathbf{s}_{\mathcal{R}}^0$ can represent v^0, ω^0, x^0, q^0, etc. Then the semi-implicit integration of rigid body dynamics is backpropagated with the chain rule as follows:

$$\frac{\mathrm{d}v^{n+1}}{\mathrm{d}s^0} = \frac{\mathrm{d}v^n}{\mathrm{d}s^0} + \Delta t \mathrm{M}^{-1} \frac{\mathrm{d}f^n}{\mathrm{d}s^0}, \tag{5}$$

$$\frac{\mathrm{d}x^{n+1}}{\mathrm{d}s^0} = \frac{\mathrm{d}x^n}{\mathrm{d}s^0} + \Delta t \frac{\mathrm{d}v^{n+1}}{\mathrm{d}s^0}, \tag{6}$$

$$\frac{\mathrm{d}\omega^{n+1}}{\mathrm{d}s^0} = \frac{\mathrm{d}\omega^n}{\mathrm{d}s^0} + \Delta t \left[\frac{\mathrm{d}(\mathrm{I}^n)^{-1}}{\mathrm{d}s^0}(L^n \times \omega^n + \tau^n) + (\mathrm{I}^n)^{-1}\left(\frac{\mathrm{d}(L^n \times \omega^n)}{\mathrm{d}s^0} + \frac{\mathrm{d}\tau^n}{\mathrm{d}s^0}\right) \right], \tag{7}$$

$$\frac{\mathrm{d}q^{n+1}}{\mathrm{d}s^0} = \frac{\mathrm{d}q^{n+1}}{\mathrm{d}\hat{q}} \left[\frac{\mathrm{d}q^n}{\mathrm{d}s^0} + \frac{\Delta t}{2} \frac{\mathrm{d}([0, \omega^{n+1}] \otimes q^n)}{\mathrm{d}s^0} \right]. \tag{8}$$

where $\hat{q} \coloneqq q^n + \frac{\Delta t}{2}([0, \omega^{n+1}] \otimes q^n)$ is the updated quaternion before the normalization operation.

Now let's dive into each gradient term in the above equations for implementation. First, we consider the gradient involving the inertia tensor I. Note that I is the function of the spatial orientation of rigid bodies since $\mathrm{I}^n = R^n \mathrm{I}^0 R^{n\top}$ with $R^n = R^n(q^n)$ (superscript n is the n^{th} time step) as the rotation matrix, so we also need to take the derivative of I^n into account. To compute $\frac{\mathrm{d}(\mathrm{I}^n)^{-1}}{\mathrm{d}s^0}(L^n \times \omega^n + \tau^n)$, we define $\Gamma \coloneqq L^n \times \omega^n + \tau^n$:

$$\begin{aligned} \frac{\mathrm{d}\,(\mathrm{I}^n)^{-1}}{\mathrm{d}s^0}\Gamma &= \frac{\mathrm{d}\left(R^n \left(\mathrm{I}^0\right)^{-1} \left(R^n\right)^\top\right)}{\mathrm{d}s^0}\Gamma \\ &= \frac{\mathrm{d}R^n}{\mathrm{d}s^0}\left(\mathrm{I}^0\right)^{-1}\left(R^n\right)^\top \Gamma + R^n\left(\mathrm{I}^0\right)^{-1}\frac{\mathrm{d}\,(R^n)^\top}{\mathrm{d}s^0}\Gamma. \end{aligned} \tag{9}$$

Since we already have $\frac{\mathrm{d}q^n}{\mathrm{d}s^0}$, to relate the gradient involving the rotational matrix $\frac{\mathrm{d}R^n}{\mathrm{d}s^0}$ with $\frac{\mathrm{d}q^n}{\mathrm{d}s^0}$, we need the following formula introduced in [Kugelstadt and Schömer 2016]:

$$\frac{\partial}{\partial q}R(q)\boldsymbol{p} = \frac{\partial}{\partial q}\left(2\boldsymbol{q}\boldsymbol{q}^\top \boldsymbol{p} + q_0^2 \boldsymbol{p} - \boldsymbol{p}\boldsymbol{q}^\top \boldsymbol{q} + 2q_0 \boldsymbol{q} \times \boldsymbol{p}\right) \tag{10}$$

where $\boldsymbol{p} \in \mathbb{R}^3$ is an arbitrary vector, and quaternion q is represented as $q = (q_0, \boldsymbol{q})$ with $q_0 \in \mathbb{R}, \boldsymbol{q} \in \mathbb{R}^3$. Then Eq. (9) can be solved as:

$$\begin{aligned} \frac{\mathrm{d}\,(\mathrm{I}^n)^{-1}}{\mathrm{d}s^0}\Gamma = &\left(\frac{\mathrm{d}R^n}{\mathrm{d}q^0}\left(\mathrm{I}^0\right)^{-1}\left(R^n\right)^\top \Gamma\right)\frac{\mathrm{d}q^n}{\mathrm{d}s^0} \\ &+ R^n\left(\mathrm{I}^0\right)^{-1}\left(\frac{\mathrm{d}\,(R^n)^\top}{\mathrm{d}q^0}\Gamma\right)\frac{\mathrm{d}q^n}{\mathrm{d}s^0}. \end{aligned} \tag{11}$$

*The corresponding author

Authors' addresses: Zhehao Li, University of Science and Technology of China, China, zhehaoli@mail.ustc.edu.cn; Qingyu Xu, University of Science and Technology of China, China, liamxu123@mail.ustc.edu.cn; Xiaohan Ye, TMCC, College of Computer Science, Nankai University, China, yexiaohan@mail.nankai.edu.cn; Bo Ren, TMCC, College of Computer Science, Nankai University, China, rb@nankai.edu.cn; Ligang Liu, University of Science and Technology of China, China, lgliu@ustc.edu.cn.

0730-0301/2023/12-ART179 $15.00
https://doi.org/10.1145/3618318

Next in Eq. (7), to compute $\frac{\mathrm{d}(L^n \times \omega^n)}{\mathrm{d}s^0}$, we have:

$$\begin{aligned}\frac{\mathrm{d}\left(\mathrm{I}^n\omega^n \times \omega^n\right)}{\mathrm{d}s^0} &= \left[\mathrm{I}^n\omega^n\right]\frac{\mathrm{d}\omega^n}{\mathrm{d}s^0} + \left[\omega^n\right]^\top \frac{\mathrm{dI}^n\omega^n}{\mathrm{d}s^0} \\ &= \left[\mathrm{I}^n\omega^n\right]\frac{\mathrm{d}\omega^n}{\mathrm{d}s^0} + \left[\omega^n\right]^\top \left(\mathrm{I}^n\frac{\mathrm{d}\omega^n}{\mathrm{d}s^0} + \frac{\mathrm{dI}^n}{\mathrm{d}s^0}\omega^n\right),\end{aligned} \tag{12}$$

where $\frac{\mathrm{dI}^n}{\mathrm{d}s^0}\omega^n$ can also be computed the same way using gradient of rotational matrix as introduced above.

Then we move to Eq. (8), $q^{n+1} = \frac{\hat{q}}{\|\hat{q}\|}$. Therefore, we have:

$$\begin{aligned}\frac{\mathrm{d}q^{n+1}}{\mathrm{d}\hat{q}} &= \frac{1}{\|\hat{q}\|}\mathbf{I}_4 - \frac{1}{\|\hat{q}\|^2}\frac{\hat{q}}{\|\hat{q}\|}\hat{q}^\top \\ &= \frac{1}{\|\hat{q}\|}\left(\mathbf{I}_4 - \hat{q}\hat{q}^\top\right),\end{aligned} \tag{13}$$

where $\mathbf{I}_4$ is the 4×4 identity matrix.

Finally in Eq. (8), to compute $\frac{\mathrm{d}([0,\omega^{n+1}]\otimes q^n)}{\mathrm{d}s^0}$, we have:

$$\begin{aligned}\frac{\mathrm{d}([0,\omega^{n+1}]\otimes q^n)}{\mathrm{d}s^0} = &\frac{\partial([0,\omega^{n+1}]\otimes q^n)}{\partial\omega^{n+1}}\frac{\mathrm{d}\omega^{n+1}}{\mathrm{d}s^0} \\ &+ \frac{\partial([0,\omega^{n+1}]\otimes q^n)}{\partial q^n}\frac{\mathrm{d}q^n}{\mathrm{d}s^0},\end{aligned} \tag{14}$$

in which we need to use the derivative of quaternion productions. As introduced in [Kugelstadt and Schömer 2016], by representing a quaternion q as $q = (q_0, \boldsymbol{q})$, the quaternion production can be written as the form of matrix product:

$$p = \hat{\boldsymbol{Q}}(q)p = \begin{pmatrix} q_0 & -\boldsymbol{q}^T \\ \boldsymbol{q} & q_0\mathbf{I}_3 - [\boldsymbol{q}]^\times \end{pmatrix}\begin{pmatrix} p_0 \\ \boldsymbol{p}\end{pmatrix}, \tag{15}$$

$$p = \boldsymbol{Q}(p)q = \begin{pmatrix} p_0 & -\boldsymbol{p}^T \\ \boldsymbol{p} & p_0\mathbf{I}_3 + [\boldsymbol{p}]^\times \end{pmatrix}\begin{pmatrix} q_0 \\ \boldsymbol{q}\end{pmatrix}, \tag{16}$$

where $\mathbf{I}_3$ is the 3×3 identity matrix. Then we can readily compute the gradient of the quaternion product using matrix calculus to obtain $\frac{\partial([0,\omega^{n+1}]\otimes q^n)}{\partial\omega^{n+1}}$ and $\frac{\partial([0,\omega^{n+1}]\otimes q^n)}{\partial q^n}$.

To sum up, we have explained each term in the gradient computation equations. However, in Eqs. (5) and (7), we need to compute the gradient of external forces f^n and torques τ^n to rigid body velocities. In a fluid-rigid coupled system, f^n, τ^n comes from the coupling forces between the fluid and the rigid body, so their gradients are introduced in the next section.

2 COMPUTATION DETAILS OF GRADIENTS OF SPH-BASED TWO-WAY FLUID-RIGID COUPLING

In this section, we introduce the computation details of the gradient of coupling forces in the SPH-based two-way coupled fluid-rigid system. We develop our differentiable two-way SPH-based fluid-rigid coupling simulator based on DFSPH [Bender and Koschier 2015, 2017] and [Akinci et al. 2012]. Here we follow the convention that for the gradient of a vector function $f \in \mathbb{R}^m$ w.r.t. a vector variable $x \in \mathbb{R}^n$, the i^{th} row of $\frac{\partial f}{\partial x} \in \mathbb{R}^{m\times n}$ is $[\frac{\partial f_i}{\partial x_1}, \frac{\partial f_i}{\partial x_2}, \cdots \frac{\partial f_i}{\partial x_n}]$, and the gradient of another vector function $g(f(x)) \in \mathbb{R}^k$ w.r.t. x is written as $\frac{\partial g}{\partial x} = \frac{\partial g}{\partial f}\frac{\partial f}{\partial x} \in \mathbb{R}^{k\times n}$.

First, let's recall the forward simulation formulas of the proposes fluid-solid coupling method in [Akinci et al. 2012], where the surfaces of rigid body $\mathcal{R}$ are sampled with boundary particles, and the coupling force and torque applied to $\mathcal{R}$ are written as double summations of rigid body particles and fluid particles:

$$f^n = \sum_{b_j}\sum_{f_i} F_{b_j\leftarrow f_i}, \quad \tau^n = \sum_{b_j}\sum_{f_i}\tau_{b_j\leftarrow f_i}, \tag{17}$$

where the pressure force $F_{b_j\leftarrow f_i}$ applied from a fluid particle f_i to a boundary particle b_j is derived as:

$$F_{b_j\leftarrow f_i} = -F_{f_i\leftarrow b_j} = m_{f_i}\Psi_{b_j}(\rho_0)\left(\frac{p_{f_i}}{\rho_0\rho_{f_i}}\right)\nabla W_{ij}, \tag{18}$$

where $\Psi_{b_j} = \rho_0 V_{b_j}$, thus the formula above can be simplified as:

$$F_{b_j\leftarrow f_i} = m_{f_i}V_{b_j}\left(\frac{p_{f_i}}{\rho_{f_i}}\right)\nabla W_{ij}. \tag{19}$$

To develop a differentiable SPH-based fluid-rigid coupling simulator, we differentiate Eq. (19) and compute $\frac{\partial F_{b_j\leftarrow f_i}}{\partial x_{b_j}}$ and $\frac{\partial F_{b_j\leftarrow f_i}}{\partial v_{b_j}}$ based on the chain rule:

$$\frac{\partial F_{b_j\leftarrow f_i}}{\partial x_{b_j}} = m_{f_i}V_{b_j}\left(\frac{1}{\rho_{f_i}}\nabla W_{ij}\frac{\partial\left(\frac{p_{f_i}}{\rho_{f_i}}\right)}{\partial x_{b_j}} + \frac{p_{f_i}}{\rho_{f_i}}\nabla^2 W_{ij}\right), \tag{20}$$

$$\frac{\partial F_{b_j\leftarrow f_i}}{\partial v_{b_j}} = m_{f_i}V_{b_j}\left(\frac{1}{\rho_{f_i}}\nabla W_{ij}\frac{\partial p_{f_i}}{\partial v_{b_j}}\right), \tag{21}$$

where in Eq. (20), the second-order spatial derivative of the cubic kernel function $\nabla^2 W$ has a closed form:

$$\nabla^2 W = \sigma_d\begin{cases} 6\left[\left(3q^2-2q\right)\frac{\mathrm{d}^2 q}{\mathrm{d}\mathbf{r}^2} + (6q-2)\frac{\mathrm{d}q}{\mathrm{d}\mathbf{r}}\left(\frac{\mathrm{d}q}{\mathrm{d}\mathbf{r}}\right)^\top\right] & \text{for } 0 \le q \le \frac{1}{2} \\ 6\left[(1-q)^2\left(-\frac{\mathrm{d}^2 q}{\mathrm{d}\mathbf{r}^2}\right) + 2(1-q)\frac{\mathrm{d}q}{\mathrm{d}\mathbf{r}}\left(\frac{\mathrm{d}q}{\mathrm{d}\mathbf{r}}\right)^\top\right] & \text{for } \frac{1}{2} < q \le 1 \\ 0 & \text{otherwise,}\end{cases} \tag{22}$$

where $q = \frac{1}{h}\|\mathbf{r}\|$, $\frac{\mathrm{d}q}{\mathrm{d}\mathbf{r}} = \frac{1}{h}\frac{\mathbf{r}}{\|\mathbf{r}\|}$, $\frac{\mathrm{d}^2 q}{\mathrm{d}\mathbf{r}^2} = \frac{1}{h\|\mathbf{r}\|}\left(I - \frac{\mathbf{r}\mathbf{r}^\top}{\|\mathbf{r}\|^2}\right)$.

Table 1. Definition of symbols

Symbol	Meaning
$\mathcal{R}$	rigid body
v^n, ω^n	linear and angular velocity of $\mathcal{R}$ at time step n
x^n, q^n	position and quaternion of $\mathcal{R}$ at time step n
s^n	state of $\mathcal{R}$ at time step n, $s^n \in \{x^n, v^n, \omega^n, q^n\}$
f^n	net fluid-rigid coupling force of $\mathcal{R}$ at time step n
τ^n	net fluid-rigid coupling torque of $\mathcal{R}$ at time step n
b_j	j^{th} rigid body particle
f_i	i^{th} fluid particle
$F_{b_j\leftarrow f_i}$	particle-pair fluid-rigid coupling force applied from f_i to b_j
ρ_0	fluid rest density
p_{f_i}, ρ_{f_i}	pressure and density of fluid particle f_i
x_{b_j}, v_{b_j}	position and velocity of rigid particle b_j
$s^n_{b_j}$	state of b_j at time step n, $s_{b_j} \in \{x_{b_j}, v_{b_j}\}$
m_{f_i}	mass of fluid particle f_i
V_{b_j}	volume of rigid particle b_j
W, W_{ij}	SPH kernel function
$k^{\text{DFSPH}}_{f_i}$	DFSPH precomputed factor of fluid particle f_i

Now let's dive into each term in Eqs. (20) and (21), To compute the gradient involving fluid particle pressure p_{f_i}, we need to investigate the pressure projection solver in the fluid simulation. Recall that in DFSPH, the pressure p_{f_i} of fluid particle f_i is solved implicitly with source term respectively being the divergence error and density error as:

$$\begin{cases} p_{f_i} = \frac{1}{\Delta t}\frac{D\rho_{f_i}}{Dt} \cdot \underbrace{\frac{\rho_{f_i}^2}{\left\|\sum_j m_j \nabla W_{ij}\right\|^2 + \sum_j \left\|m_j \nabla W_{ij}\right\|^2}}_{k_{f_i}^{\text{DFSPH}}}, \\ p_{f_i} = \frac{1}{\Delta t^2}\left(\rho_{f_i}^* - \rho_0\right) k_{f_i}^{\text{DFSPH}}, \end{cases} \tag{23}$$

where m_{f_i}, ρ_{f_i} is the mass and density of f_i; $W_{ij} = W(x_i - x_j, h)$ is the kernel function with support radius h and two particle positions x_i, x_j; $\frac{D\rho_{f_i}}{Dt} = \sum_j m_j \left(\mathbf{v}_i - \mathbf{v}_j\right) \cdot \nabla W_{ij}$ is the density error caused by velocity advection and $\rho_{f_i}^* = \rho_{f_i} + \Delta t \frac{D\rho_{f_i}}{Dt}$ is the predicted density. In practice, the ρ_{f_i} in $k_{f_i}^{\text{DFSPH}}$ is approximated with ρ_0 thus can be moved into the computation of $\frac{D\rho_{f_i}}{Dt}$ and $\rho_{f_i}^* - \rho_0$ to simplify the computation, then the simplified $\hat{k}_{f_i}^{\text{DFSPH}}$ is computed as:

$$\hat{k}_{f_i}^{\text{DFSPH}} = \frac{1}{\left\|\sum_j m_j \nabla W_{ij}\right\|^2 + \sum_j \left\|m_j \nabla W_{ij}\right\|^2}. \tag{24}$$

To compute the gradient, we directly differentiate Eq. (23). In the density error solver, we compute the gradient based on chain rules with $s_{b_j} \in (x_{b_j}, v_{b_j})$:

$$\frac{\partial p_{f_i}}{\partial s_{b_j}} = \frac{1}{\Delta t^2}\left(\frac{\partial \rho_{f_i}^*}{\partial s_{b_j}}\hat{k}_{f_i}^{\text{DFSPH}} + (\rho_{f_i}^* - \rho^0)\frac{\partial \hat{k}_{f_i}^{\text{DFSPH}}}{\partial s_{b_j}}\right). \tag{25}$$

In the divergence error solver, we compute the gradient the same way, since the computational structure of the divergence error solver and density error solver is the same but with different source terms:

$$\frac{\partial p_{f_i}}{\partial s_{b_j}} = \frac{1}{\Delta t}\left(\frac{\partial\left(\frac{D\rho_{f_i}}{Dt}\right)}{\partial s_{b_j}}\hat{k}_{f_i}^{\text{DFSPH}} + \frac{D\rho_{f_i}}{Dt}\frac{\partial \hat{k}_{f_i}^{\text{DFSPH}}}{\partial s_{b_j}}\right), \tag{26}$$

In which $\frac{\partial\left(\frac{D\rho_{f_i}}{Dt}\right)}{\partial s_{b_j}}$ is computed as:

$$\frac{\partial\left(\frac{D\rho_{f_i}}{Dt}\right)}{\partial v_{b_j}} = -m_{b_j}\nabla W_{ij}, \quad \frac{\partial\left(\frac{D\rho_{f_i}}{Dt}\right)}{\partial x_{b_j}} = m_{b_j}\nabla^2 W_{ij}(v_{f_i} - v_{b_j}). \tag{27}$$

For $\frac{\partial \hat{k}_{f_i}^{\text{DFSPH}}}{\partial s_{b_j}}$ and $\frac{\partial \hat{k}_{f_i}^{\text{DFSPH}}}{\partial v_{b_j}}$, since v_{b_j} is not directly involved in the comuputation of $\hat{k}_{f_i}^{\text{DFSPH}}$, $\frac{\partial \hat{k}_{f_i}^{\text{DFSPH}}}{\partial v_{b_j}} = \mathbf{0}$, and $\frac{\partial \hat{k}_{f_i}^{\text{DFSPH}}}{\partial x_{b_j}}$ is computed as:

$$\frac{\partial \hat{k}_{f_i}^{\text{DFSPH}}}{\partial x_{b_j}} = \alpha\left[\left(m_{b_j}\nabla^2 W_{ij}\right)\left(\sum_j m_j \nabla W_{ij}\right) + m_{b_j}\nabla^2 W_{ij}\nabla W_{ij}\right], \tag{28}$$

ALGORITHM 1: Constant density solver with gradient computation

```
Function DensitySolverWithGradient(k^DFSPH)
    ε = density error
    while (ε > thresh) do
        compute ρ_fi for each fluid particle
        // compute gradient before updating v_f:
        parallel forall dynamic boundary rigid body particle b_j do
            sequential forall neighbor fluid particles f_i do
                compute ∂F_{b_j←f_i}/∂v_{b_j}, ∂F_{b_j←f_i}/∂v_{f_i} with Eqs. (21)(25)(29), etc.
                compute dF_{b_j←f_i}/dv_{b_j} = (I + Δt/m_{f_i} ∂F_{b_j←f_i}/∂v_{f_i})^{-1} ∂F_{b_j←f_i}/∂v_{b_j}
            accumulate & store Σ_{f_i} ∂F_{b_j←f_i}/∂s^n = Σ_{f_i} dF_{b_j←f_i}/ds_{b_j} ∂s_{b_j}/∂s^n
        // original forward simulation part:
        parallel forall fluid particle f do
            traverse all neighbor fluid & rigid particles to compute
              pressure forces to update v_f^* (Alg. (1) in main paper)
    end
    return updated fluid particle velocities v_f^*
end
```

where $\alpha := \frac{-2}{\left(\left\|\sum_j m_j \nabla W_{ij}\right\|^2 + \sum_j \left\|m_j \nabla W_{ij}\right\|^2\right)^2}$. With these formulas, now we can compute the gradient of fluid particle pressure w.r.t. rigid body particle state.

To sum up, with all the derived Eqs. (20)(21)(25)(26), we are able to combine them to compute $\frac{\partial F_{b_j \leftarrow f_i}}{\partial x_{b_j}}$ and $\frac{\partial F_{b_j \leftarrow f_i}}{\partial v_{b_j}}$. It is worth noting that the above computation process seems complicated, but as analyzed in the main paper with respect to the gradient instability issue, in the final proposed localized gradient computational scheme, we only need to compute $\frac{\partial F_{b_j \leftarrow f_i}}{\partial v_{b_j}}$. $\frac{\partial F_{b_j \leftarrow f_i}}{\partial v_{f_i}}$ can be computed in a similar way by replacing the role of v_{b_j} in Eqs. (21)(25)(26) with v_{f_i}. For the convenience of our readers, we include the formula for the density solver below (divergence solver in a similar way):

$$\begin{aligned} \frac{\partial F_{b_j \leftarrow f_i}}{\partial v_{f_i}} &= m_{f_i} V_{b_j}\left(\frac{1}{\rho_{f_i}}\nabla W_{ij}\frac{\partial p_{f_i}}{\partial v_{f_i}}\right) \\ &= m_{f_i} V_{b_j}\left(\frac{1}{\rho_{f_i}}\nabla W_{ij}\frac{1}{\Delta t^2}\hat{k}_{f_i}^{\text{DFSPH}}\frac{\partial \rho_{f_i}^*}{\partial v_{f_i}}\right) \\ &= m_{f_i} V_{b_j}\left(\frac{1}{\rho_{f_i}}\nabla W_{ij}\frac{1}{\Delta t^2}\hat{k}_{f_i}^{\text{DFSPH}}\left(\Delta t\sum_j m_j \nabla W_{ij}\right)\right). \end{aligned} \tag{29}$$

.

To help readers better understand the process, we include the algorithm for the density solver with gradient computation as Alg. (1), and the divergence solver with gradient follows the same structure.

Finally, our goal is to get the final gradient of fluid-rigid coupling forces and torques with respect to the rigid body state s^n by accumulating the gradient of each rigid body particle as:

$$\frac{\partial f^n}{\partial s^n} = \sum_{b_j}\sum_{f_i}\frac{\partial F_{b_j \leftarrow f_i}}{\partial s^n} = \sum_{b_j}\sum_{f_i}\frac{\mathrm{d}F_{b_j \leftarrow f_i}}{\mathrm{d}s_{b_j}}\frac{\partial s_{b_j}}{\partial s^n}, \tag{30}$$

where the formula of $\frac{\partial \tau^n}{\partial s^n}$ has the same structure. However, there is a gap between $\frac{\mathrm{d}F_{b_j \leftarrow f_i}}{\mathrm{d}\mathbf{s}_{b_j}}$ in Eq. (30) and $\frac{\partial F_{b_j \leftarrow f_i}}{\partial \mathbf{s}_{b_j}}$ in Eqs. (20)(21), which means there are still other terms that need to be taken into account in gradient computation. This is related to how we decide to differentiate the iterative solving process of DFSPH pressure solvers, and we will introduce the computation details of the general while unstable gradient computation scheme presented in the main paper in the next section.

3 COMPUTATION DETAILS OF THE GENERAL BUT UNSTABLE GRADIENT COMPUTATION SCHEME

In this section, we introduce the computational details of the general but unstable gradient computation scheme of SPH-based two-way fluid-rigid coupling presented in the main paper.

First, by considering the neighboring fluid particles of rigid body particles in DFSPH pressure projection, it can be deduced from the conclusions in the main paper that:

$$\begin{aligned}\frac{\mathrm{d}F_{b_j \leftarrow f_i}}{\mathrm{d}\mathbf{s}_{b_j}} &= \frac{\partial F_{b_j \leftarrow f_i}}{\partial \mathbf{s}_{b_j}} + \frac{\partial F_{b_j \leftarrow f_i}}{\partial v_{f_i}} \frac{\mathrm{d}v_{f_i}}{\mathrm{d}\mathbf{s}_{b_j}} + \sum_{f_j} \frac{\partial F_{b_j \leftarrow f_i}}{\partial v_{f_j}} \frac{\mathrm{d}v_{f_j}}{\mathrm{d}\mathbf{s}_{b_j}} \\ &= \frac{\partial F_{b_j \leftarrow f_i}}{\partial \mathbf{s}_{b_j}} + \frac{\partial F_{b_j \leftarrow f_i}}{\partial v_{f_i}} \frac{\mathrm{d}v_{f_i}}{\mathrm{d}\mathbf{s}_{b_j}} \\ &\quad + \sum_{f_j} \frac{\partial F_{b_j \leftarrow f_i}}{\partial v_{f_j}} \left(\frac{\partial v_{f_j}}{\partial \mathbf{s}_{b_j}} + \sum_{f_p} \frac{\partial v_{f_j}}{\partial v_{f_p}} \frac{\mathrm{d}v_{f_p}}{\mathrm{d}\mathbf{s}_{b_j}} \right),\end{aligned} \tag{31}$$

where f_p is the neighboring particle of f_j, and we can further expand $\frac{\mathrm{d}v_{f_p}}{\mathrm{d}\mathbf{s}_{b_j}}$ to a formula contain $\frac{\mathrm{d}v_{f_i}}{\mathrm{d}\mathbf{s}_{b_j}}$. With the relationship between particle accelerations and coupling forces, as the iterative solving process goes on, k^{th}-order neighboring fluid particles are taken into account, leading to a recursive gradient computation formulation.

We can rewrite this formula to the gradient of $F_{b_j \leftarrow f_i}$ w.r.t. rigid body state s^n as:

$$\frac{\mathrm{d}F_{b_j \leftarrow f_i}}{\mathrm{d}\mathbf{s}^n} = \frac{\partial F_{b_j \leftarrow f_i}}{\partial \mathbf{s}^n} + \frac{\partial F_{b_j \leftarrow f_i}}{\partial v_{f_i}} \frac{\mathrm{d}v_{f_i}}{\mathrm{d}\mathbf{s}^n} + \sum_{f_p} \frac{\partial F_{b_j \leftarrow f_i}}{\partial v_{f_p}} \frac{\mathrm{d}v_{f_p}}{\mathrm{d}\mathbf{s}^n}, \tag{32}$$

where f_p represents the fluid particle other than the first-ring neighbor fluid particles f_i of b_j.

To compute Eq. (32), the full velocity derivatives of each fluid particle $\frac{\mathrm{d}v_{f_i}}{\mathrm{d}\mathbf{s}^n}$, $\frac{\mathrm{d}v_{f_p}}{\mathrm{d}\mathbf{s}^n}$ are needed. Now we consider the relationship between fluid-rigid coupling force and velocity of fluid particles. In each iteration step of pressure projection, the fluid particle velocities update as (with superscript k denotes the k^{th} iteration):

$$v_{f_i}^{(k+1)} = v_{f_i}^{(k)} + \frac{\Delta t}{m_{f_i}} \sum_{b_j} F_{f_i \leftarrow b_j}^{(k)} + \frac{\Delta t}{m_{f_i}} \sum_{f_j} F_{f_i \leftarrow f_j}^{(k)}. \tag{33}$$

Therefore, gradients to the rigid body status could be derived iteratively by combining Eq. (32)(33):

$$\begin{aligned}\frac{\mathrm{d}v_{f_i}^{(k+1)}}{\mathrm{d}\mathbf{s}^n} &= \frac{\mathrm{d}v_{f_i}^{(k)}}{\mathrm{d}\mathbf{s}^n} \\ &\quad + \frac{\Delta t}{m_{f_i}} \sum_{b_j} \left(\frac{\partial F_{f_i \leftarrow b_j}^{(k)}}{\partial \mathbf{s}^n} + \frac{\partial F_{f_i \leftarrow b_j}^{(k)}}{\partial v_{f_i}^{(k)}} \frac{\mathrm{d}v_{f_i}^{(k)}}{\mathrm{d}\mathbf{s}^n} + \sum_{f_p} \frac{\partial F_{f_i \leftarrow b_j}^{(k)}}{\partial v_{f_p}^{(k)}} \frac{\mathrm{d}v_{f_p}^{(k)}}{\mathrm{d}\mathbf{s}^n} \right) \\ &\quad + \frac{\Delta t}{m_{f_i}} \sum_{f_j} \left(\frac{\partial F_{f_i \leftarrow f_j}^{(k)}}{\partial \mathbf{s}^n} + \frac{\partial F_{f_i \leftarrow f_j}^{(k)}}{\partial v_{f_i}^{(k)}} \frac{\mathrm{d}v_{f_i}^{(k)}}{\mathrm{d}\mathbf{s}^n} + \sum_{f_p} \frac{\partial F_{f_i \leftarrow f_j}^{(k)}}{\partial v_{f_p}^{(k)}} \frac{\mathrm{d}v_{f_p}^{(k)}}{\mathrm{d}\mathbf{s}^n} \right).\end{aligned} \tag{34}$$

Then, to form a recursive solving formula, we obtain the gradient of fluid-rigid forces by replacing $\frac{\mathrm{d}v_{f_i}}{\mathrm{d}\mathbf{s}^n}$, $\frac{\mathrm{d}v_{f_p}}{\mathrm{d}\mathbf{s}^n}$ in Eq. (32) with $\frac{\mathrm{d}v_{f_i}^{(k)}}{\mathrm{d}\mathbf{s}^n}$, $\frac{\mathrm{d}v_{f_p}^{(k)}}{\mathrm{d}\mathbf{s}^n}$, as:

$$\frac{\mathrm{d}F_{f_i \leftarrow b_j}^{(k)}}{\mathrm{d}\mathbf{s}^n} = \frac{\partial F_{f_i \leftarrow b_j}^{(k)}}{\partial \mathbf{s}^n} + \frac{\partial F_{f_i \leftarrow b_j}^{(k)}}{\partial v_{f_i}^{(k)}} \frac{\mathrm{d}v_{f_i}^{(k)}}{\mathrm{d}\mathbf{s}^n} + \sum_{f_p} \frac{\partial F_{f_i \leftarrow b_j}^{(k)}}{\partial v_{f_p}^{(k)}} \frac{\mathrm{d}v_{f_p}^{(k)}}{\mathrm{d}\mathbf{s}^n}, \tag{35}$$

where we assume that for all fluid particle f, at the beginning of the iteration $\frac{\mathrm{d}v_f}{\mathrm{d}\mathbf{s}_{b_j}} = \mathbf{0}$.

Finally, since the fluid-rigid coupling forces are simply accumulated along the k times iterations in a single time step of DFSPH pressure projection, the total force gradients can be derived by summation:

$$\frac{\mathrm{d}F_{f_i \leftarrow b_j}}{\mathrm{d}\mathbf{s}^n} = \sum \frac{\mathrm{d}F_{f_i \leftarrow b_j}^{(k)}}{\mathrm{d}\mathbf{s}^n}. \tag{36}$$

To sum up, as iteration index k increases, to update the velocity v_{f_i} of fluid particle f_i, k^{th}-order neighboring fluid particles are involved, then the recursive solving of the fluid-rigid coupling gradient gradually accumulates the gradient contribution from the k^{th}-order of fluid particles. However, we find that in practice this general gradient computation scheme encounters quick gradient explosion issues, as stated in the main paper.

4 COMPUTATIONAL DETAILS OF DIFFERENTIABLE SPH-BASED RIGID-RIGID CONTACT

As shown in the main paper, we present a penalty-based SPH-based rigid-rigid contact model for its good differentiability.

$$F_r^{\text{normal}} = k(\frac{\rho_r}{\rho_0} - 1)\mathbf{n}_\mathbf{r}, \quad F_r^{\text{friction}} = -\mu \|F^{\text{normal}}\| \frac{\mathbf{v}_\mathbf{r}^{\text{rel}}}{\|\mathbf{v}_\mathbf{r}^{\text{rel}}\|}, \tag{37}$$

The gradient of the normal force to rigid particle density is:

$$\frac{\mathrm{d}F_r^{\text{normal}}}{\mathrm{d}\rho_r} = \frac{k}{\rho_0}\mathbf{n}_r, \quad \frac{\mathrm{d}F_r^{\text{friction}}}{\mathrm{d}\rho_r} = -\mu \frac{\mathbf{v}_r^{\text{rel}}}{\|\mathbf{v}_r^{\text{rel}}\|} \left(\frac{F_r^{\text{normal}}}{\|F_r^{\text{normal}}\|} \right)^\top \frac{\mathrm{d}F_r^{\text{normal}}}{\mathrm{d}\rho_r} \tag{38}$$

Given the gradient of the density of a rigid particle to its positions or another rigid particle $\frac{\mathrm{d}\rho_r}{\mathrm{d}x}$(which can be easily computed by differentiating the kernel function in SPH density formulation), we can compute the rigid contact normal force and friction force to the rigid particles in contact. The friction force to particle velocity can also be easily computed by differentiating $\mathbf{v}_r^{\text{rel}}$. Then we accumulate the gradient of all rigid particles to get the final gradient of contact forces to rigid body state.

Table 2. Summary of the optimization parameters of the gradient descent optimizer and scene settings used in our rigid body trajectory optimization experiments. lr_v^0, lr_ω^0 denotes the initial learning rate for linear and angular velocities, w_{position}, w_{rotation} are the weight factor for position and rotation energy, and patience is the patience parameter for the learning rate scheduler. x^0, x^*, θ^*, t^* denotes the initial position, the target position, the target rotation angle on the XYZ axis of the rigid body (In practice we use quaternion for orientation), and the target time respectively. "-" means not to optimize this variable in the experiment.

Task	lr_v^0	lr_ω^0	w_{position}	w_{rotation}	patience	x^0	x^*	θ^*	t^*
Water Bottle Flip	1	1	0.1	1	5	(-2, 1.5, 0)	(4.3, 3.5, 0)	(X 0°, Y 0°, Z 360°)	1.0s
Stone Skipping	8	1	1	-	5	(-2, 1.2, 0)	(1.7, 1.6, 0)	-	0.23s
Water Rafting	0.1	0.1	1	1	10	(-1.0, 0.6, -0.6)	(1.0, 0.5, 0.7)	(X 0°, Y 180°, Z 0°)	2.0s
High Diving	-	1	-	1	20	-	-	(X 0°, Y 180°, Z 0°)	2.3s
On-water Billiards	0.2	0.2	1	-	5	(-0.1, 0.46, 0)	(0.5, 0.5, -0.5)	-	0.25s

5 COMPUTATION DETAILS OF GRADIENTS OF IMPULSE-BASED MULTI-BODY SIMULATION

In the example of the 2D on-water inverted pendulum robot, we use a spherical joint to connect the pole and the cart, and adopt the impulse-based multi-body simulation method from [Bender and Schmitt 2006]. In this method, the constraints in articulated systems are satisfied by introducing additional impulses to the integration of each part of the multi-body system, which we find in practice can be more easily integrated into our framework compared with other articulated system simulation methods.

In the spherical joint, the constraint is that the distance d between the joint points x_1, x_2 of two bodies should always be 0. In the forward simulation, suppose we do not consider this constraint in one time step with initial time t_0, then at the end of the time step, d becomes:

$$d(t_0 + \Delta t) = x_1(t_0 + \Delta t) - x_2(t_0 + \Delta t). \tag{39}$$

To make $d(t_0+\Delta t)$ still equal to 0, the idea of [Bender and Schmitt 2006] is to introduce an additional impulse into the system to correct this distance error at the end of this time step. To compute this impulse, we first introduce the change of velocity of one point on a rigid body in the impact of external impulse. Let $P(t)$ and $Q(t)$ be two arbitrary points of the i-th rigid body in world space and let $r_P(t) = P(t) - C_i(t)$ and $r_Q(t) = Q(t) - C_i(t)$ be the vectors from the mass center C_i to these points. If an impulse p is applied at $Q(t)$, the change $\Delta v_P(t)$ of the point velocity of $P(t)$ can be computed with the following matrix $K_{P,Q}(t)$:

$$\begin{aligned} K_{P,Q}(t) &:= \begin{cases} \frac{1}{m_i}\mathbf{I}_3 - [r_P(t)]\,\mathrm{I}_i^{-1}(t)\,[r_Q(t)] & \text{if body } i \text{ is dynamic} \\ \mathbf{0} & \text{otherwise} \end{cases} \\ \Delta v_P(t) &= K_{P,Q}(t) \cdot p, \end{aligned} \tag{40}$$

where $\mathbf{I}_3$ is the 3×3 identity matrix and m_i, I_i is the mass and the inertia tensor of the i-th rigid body. In the on-water inverted pendulum robot example, suppose the distance between the pivot point and the mass centers of the pole and the cart is $R_1 l_1, R_2 l_2$ with R be the rotational matrix, then K is computed as:

$$K_i = \frac{1}{m_i}\mathbf{I}_3 - [R_i l_i]\,\mathrm{I}_i^{-1}(t)\,[R_i l_i], \quad i = 1, 2 \tag{41}$$

Then we can compute the impulse p needed in the spherical joint by solving the following equation:

$$K_1(t_0) \cdot p - K_2(t_0) \cdot (-p) = \frac{1}{\Delta t} d(t_0 + \Delta t), \tag{42}$$

The matrix $K(t_0) := K_1(t_0) + K_2(t_0)$ is constant at time t_0 and is nonsingular, symmetric and position definite (proof in [Mirtich 1996]). Therefore, the equation above can be solved by inverting $K(t_0)$:

$$p = \frac{1}{\Delta t} K(t_0)^{-1} d(t_0 + \Delta t). \tag{43}$$

To differentiate the impulse-based multi-body simulation in the on-water inverted pendulum robot example, we need to compute the gradient of this impulse to the velocity of the cart, which can be done based on the chain rule:

$$\frac{\mathrm{d}p}{\mathrm{d}v} = \frac{1}{\Delta t}\left(\frac{\mathrm{d}K^{-1}}{\mathrm{d}v} d + K^{-1}\frac{\mathrm{d}d}{\mathrm{d}v}\right), \tag{44}$$

where $\frac{\mathrm{d}K^{-1}}{\mathrm{d}v}$ and $\frac{\mathrm{d}d}{\mathrm{d}v}$ are computed as:

$$\frac{\mathrm{d}K^{-1}}{\mathrm{d}v} d = -K^{-1}\frac{\mathrm{d}K}{\mathrm{d}v}K^{-1} d \tag{45}$$

$$\frac{\mathrm{d}K}{\mathrm{d}v} d = -\frac{\mathrm{d}\left(\sum_{i=1}^{2}[R_i l_i]\,\mathrm{I}_i^{-1}(t)\,[R_i l_i]\right)}{\mathrm{d}v} d, \tag{46}$$

$$\frac{\mathrm{d}d}{\mathrm{d}v} = \frac{\mathrm{d}(x_1 - x_2)}{\mathrm{d}v} + \frac{\mathrm{d}(R_1 l_1 - R_2 l_2)}{\mathrm{d}v}, \tag{47}$$

which can be computed with the techniques to compute gradient involving rotational matrix introduced in Sec. 1. Finally, we integrate the gradient of this additional impulse into our framework to extend our differentiable particle-based fluid-rigid coupling simulator to support multi-body systems.

6 EXPERIMENT DETAILS

6.1 Experiment parameters

We summarize the optimization parameters and scene settings used in the rigid body trajectory optimization experiments in Tab. 2.

6.2 Influence of Initial Learning Rate

We evaluate the influence of the initial learning rate of the gradient descent method on the results. In practice, we only manually set the initial learning rate and let the learning rate scheduler automatically reduce the learning rate once by half based on optimization performance. Our choice of initial learning rate is based on a simple strategy: when we observe overshooting, we lower the initial learning rate by half otherwise we increase the initial learning rate. We choose the water bottle flip task as an example to show the influence of the initial learning rates on the optimization results.

The results are shown in Fig. 1, where the positional and rotational energy share the same initial learning rate. In this task, the results are not very sensitive to the initial learning rate, and similar results are observed in other tasks.

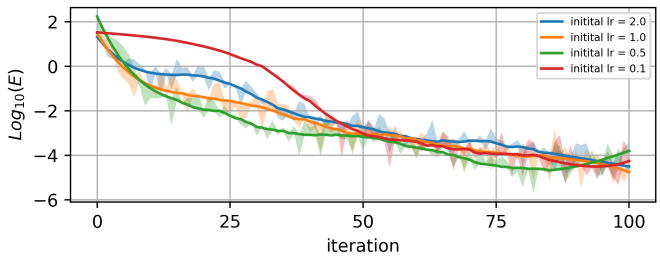

Fig. 1. Optimization results of the gradient descent optimizer on the water bottle flipping task with different initial learning rates.

REFERENCES

Nadir Akinci, Markus Ihmsen, Gizem Akinci, Barbara Solenthaler, and Matthias Teschner. 2012. Versatile rigid-fluid coupling for incompressible SPH. *ACM Trans. Graph.* 31, 4 (Aug. 2012), 1–8. https://doi.org/10.1145/2185520.2185558

Jan Bender and Dan Koschier. 2015. Divergence-free smoothed particle hydrodynamics. In *Proceedings of the 14th ACM SIGGRAPH / Eurographics Symposium on Computer Animation*. ACM, Los Angeles California, 147–155. https://doi.org/10.1145/2786784.2786796

Jan Bender and Dan Koschier. 2017. Divergence-Free SPH for Incompressible and Viscous Fluids. *IEEE Trans. Visual. Comput. Graphics* 23, 3 (March 2017), 1193–1206. https://doi.org/10.1109/TVCG.2016.2578335

Jan Bender and Alfred Schmitt. 2006. Fast Dynamic Simulation of Multi-Body Systems Using Impulses. In *Virtual Reality Interactions and Physical Simulations (VRIPhys)*. Madrid (Spain), 81–90.

Tassilo Kugelstadt and Elmar Schömer. 2016. Position and Orientation Based Cosserat Rods. In *Eurographics/ ACM SIGGRAPH Symposium on Computer Animation*, Ladislav Kavan and Chris Wojtan (Eds.). The Eurographics Association. https://doi.org/10.2312/sca.20161234

Brian Vincent Mirtich. 1996. *Impulse-Based Dynamic Simulation of Rigid Body Systems*. Ph. D. Dissertation. AAI9723116.