EUROGRAPHICS 2024 / A. Bermano and E. Kalogerakis
(Guest Editors)

COMPUTER GRAPHICS *forum*
Volume 43 (2024), Number 2

Appendix: The Impulse Particle-In-Cell Method

Sergio Sancho[1,2], Jingwei Tang[2], Christopher Batty[3], Vinicius C. Azevedo[2]

[1]ETH Zürich, Switzerland [2]DisneyResearch|Studios, Switzerland [3]University of Waterloo, Canada

1. Derivation of the Impulse Equation

This section provides a Lagrangian derivation for the impulse equation of motion based on the material derivative $\frac{D}{Dt} = \frac{\partial}{\partial t} + \mathbf{u} \cdot \nabla$, following [Cor95]. As discussed in the paper, the impulse variable is defined as

$$\mathbf{m} = \mathbf{u} + \nabla\phi, \tag{1}$$

where ϕ is any scalar field and we have $\nabla \cdot \mathbf{u} = 0$. If we take the curl in Equation 1 we recover the exact vorticity $\omega = \nabla \times \mathbf{m} = \nabla \times \mathbf{u}$. Note that by taking the divergence $\nabla \cdot \mathbf{m} = \nabla^2\phi$ we obtain a Poisson equation that allows us to recover the divergence-free velocity $\mathbf{u}$ through a projection.

For the next derivation, consider the following identities:

$$\nabla\left(\frac{1}{2}|\mathbf{u}|^2\right) = (\nabla\mathbf{u})^T\,\mathbf{u} \tag{ID.1}$$

$$\frac{D}{Dt}\nabla\phi = \nabla\left(\frac{D\phi}{Dt}\right) - (\nabla\mathbf{u})^T\,\nabla\phi \tag{ID.2}$$

Starting from Equation 1, we can apply the material derivative on both sides of the equation and use the identities **ID.1** and **ID.2** to obtain the following result:

$$\begin{aligned}\frac{D\mathbf{m}}{Dt} &= \frac{D\mathbf{u}}{Dt} + \frac{D}{Dt}\nabla\phi = \frac{D\mathbf{u}}{Dt} + \nabla\left(\frac{D\phi}{Dt}\right) - (\nabla\mathbf{u})^T\,\nabla\phi \\ &= \frac{D\mathbf{u}}{Dt} + \nabla\left(\frac{D\phi}{Dt} + \frac{1}{2}|\mathbf{u}|^2\right) - (\nabla\mathbf{u})^T\,\mathbf{m}\end{aligned} \tag{2}$$

In the last equation, we know that for an inviscid and incompressible fluid $\frac{D\mathbf{u}}{Dt} = -\nabla p$. If we substitute this identity in Equation 2 we obtain

$$\frac{D\mathbf{m}}{Dt} = \nabla\left(\frac{D\phi}{Dt} + \frac{1}{2}|\mathbf{u}|^2 - p\right) - (\nabla\mathbf{u})^T\,\mathbf{m}. \tag{3}$$

Additionally, the last expression depends on an initial definition of ϕ. Since this scalar field can be arbitrarily chosen, we can conveniently let ϕ evolve according to

$$\frac{D\phi}{Dt} + \frac{1}{2}|\mathbf{u}|^2 - p = 0, \tag{4}$$

which implies that the gradient in Equation 3 cancels out and reduces this expression to our equation of motion for the impulse variable:

$$\frac{\partial\mathbf{m}}{\partial t} + \mathbf{u} \cdot \nabla\mathbf{m} + (\nabla\mathbf{u})^T\,\mathbf{m} = 0 \tag{5}$$

2. Derivation of the Velocity Jacobian Correction

We can compute the Jacobian of the flow map per particle and apply the stretching correction on the velocity that the particle is carrying during the APIC advection. Since we are following the impulse equation, we will refer to the stretched quantity as impulse denoting it by $\mathbf{m}$. We can modify the per-particle stored impulse as

$$\mathbf{m}^\circ = (\nabla\Phi^\circ)^{-T}\mathbf{u}^\circ. \tag{6}$$

The problem of correcting the carried impulse by a matrix in the particle space is that APIC also utilizes the affine term $\mathbf{A}^\circ = \nabla\mathbf{u}^\circ$ during the particle-to-grid transfer, so a proper modification of this term should be performed. Even though the flow map can be assumed to be locally constant in space to get a simpler approach, we can provide a more accurate derivation that respects the space-dependency of the flow map function. Our goal is to obtain the APIC affine term for the new modified impulse, i.e., $\mathbf{A}^\circ_\mathbf{m} = \nabla\mathbf{m}^\circ$.

For the rest of this section, the flow map is a function of $\mathbf{x}$, that is, $\Phi(\mathbf{x})$, so the corresponding Jacobian matrix is also dependent of $\mathbf{x}$. We will drop the particle/grid superscript symbols to provide a cleaner derivation. For convenience and to avoid the appearance of third order tensors, we will derive the matrix-vector product $\nabla\mathbf{m}\,\Delta\mathbf{x}$, with the constant vector $\Delta\mathbf{x} = \mathbf{x}^\boxplus - \mathbf{x}^\circ_p$, since this is the only operation where we need to apply the impulse APIC term. In terms of notation, we will refer to the transpose Jacobian matrix of the flow map $(\nabla\Phi)^T$ as a generic invertible matrix $\mathbf{F}(\mathbf{x})$ that depends on $\mathbf{x}$, which can be written in the form

$$\mathbf{F}(\mathbf{x}) = \sum_{ij} f_{ij}(\mathbf{x})\,\mathbf{E}_{ij}$$

where $\mathbf{E}_{ij}$ has a 1 in the (i,j) position and zeros in all other positions.

We have the expression

$$\nabla\mathbf{m}\,\Delta\mathbf{x} = \nabla\left(\mathbf{F}^{-1}(\mathbf{x})\mathbf{u}(\mathbf{x})\right)\Delta\mathbf{x}$$

that can be extended applying the product rule to

$$\nabla\mathbf{m}\,\Delta\mathbf{x} = \mathbf{F}^{-1}(\mathbf{x})\nabla\mathbf{u}(\mathbf{x})\Delta\mathbf{x} + \nabla\left(\mathbf{F}^{-1}(\mathbf{x})\hat{\mathbf{u}}\right)\Delta\mathbf{x},$$

where $\hat{\mathbf{u}}$ is the vector $\mathbf{u}(\mathbf{x})$ treated as a constant vector. Note that the last expression avoids the use of third order tensors. We can notice that the last term can be treated as an extension of a directional derivative for a vector-valued function $\mathbf{F}^{-1}(\mathbf{x})\hat{\mathbf{u}}$ in the direction and magnitude of $\Delta\mathbf{x}$ and evaluated at $\mathbf{x}$, which we will denote as $\nabla_{\Delta\mathbf{x}}\left[\mathbf{F}^{-1}(\mathbf{x})\,\hat{\mathbf{u}}\right]$. Thus, we have

$$\nabla_{\Delta\mathbf{x}}\left[\mathbf{F}^{-1}(\mathbf{x})\hat{\mathbf{u}}\right] = -\mathbf{F}^{-1}(\mathbf{x})\;\nabla_{\Delta\mathbf{x}}\mathbf{F}(\mathbf{x})\;\mathbf{F}^{-1}(\mathbf{x})\;\hat{\mathbf{u}},$$

where matrix $\nabla_{\Delta\mathbf{x}}\mathbf{F}(\mathbf{x})$ contains the derivatives of $\mathbf{F}$ and is defined as

$$\begin{aligned}\nabla_{\Delta\mathbf{x}}\mathbf{F}(\mathbf{x}) &= \nabla_{\Delta\mathbf{x}}\sum_{ij} f_{ij}(\mathbf{x})\mathbf{E}_{ij} = \sum_{ij}\nabla_{\Delta\mathbf{x}} f_{ij}(\mathbf{x})\,\mathbf{E}_{ij}\\ &= \sum_{ij}\left(\nabla f_{ij}(\mathbf{x})^{T}\Delta\mathbf{x}\right)\mathbf{E}_{ij},\end{aligned}$$

and we have used the matrix rule $\partial\left(\mathbf{X}^{-1}\right) = -\mathbf{X}^{-1}\,(\partial\mathbf{X})\,\mathbf{X}^{-1}$.

We can now obtain the final expression

$$\nabla\mathbf{m}\,\Delta\mathbf{x} = \mathbf{F}^{-1}(\mathbf{x})\nabla\mathbf{u}(\mathbf{x})\Delta\mathbf{x}\;-\mathbf{F}^{-1}(\mathbf{x})\;\nabla_{\Delta\mathbf{x}}\mathbf{F}(\mathbf{x})\;\mathbf{F}^{-1}(\mathbf{x})\;\hat{\mathbf{u}}.$$

The last equations shows how to compute the new affine term $\mathbf{A}^{\circ}_{\mathbf{m}} = \nabla\mathbf{m}^{\circ}$ associated with the corrected $\mathbf{m}^{\circ}$ that we need in order to perform an accurate APIC transfer. Recovering $\mathbf{F}(\mathbf{x}) = (\nabla\Phi^{\circ})^{\top}$, we get the following expression

$$\mathbf{A}^{\circ}_{\mathbf{m}}\,\Delta\mathbf{x} = \mathcal{J}^{\circ}\mathbf{A}^{\circ}\Delta\mathbf{x}\;-\mathcal{J}^{\circ}\;\nabla_{\Delta\mathbf{x}}\left[(\nabla\Phi^{\circ})^{\top}\right]\;\mathcal{J}^{\circ}\;\mathbf{u}^{\circ}.$$

The matrix $\nabla_{\Delta\mathbf{x}}\left[(\nabla\Phi^{\circ})^{-\top}\right]$ contains the second derivative terms that come from the Hessian tensor of the flow map. Despite having included the vector $\Delta\mathbf{x}$ in the derivation for convenience, we would like to extract it as an independent vector to be able to reuse the existing infrastructure of an APIC transfer. Rearranging the terms easily leads us to the following expression:

$$\mathbf{A}^{\circ}_{\mathbf{m}}\,\Delta\mathbf{x} = \mathcal{J}^{\circ}(\mathbf{A}^{\circ} - \mathbf{H}^{\circ})\,\Delta\mathbf{x} \tag{7}$$

where the matrix $\mathbf{H}^{\circ}$ can be computed in 2D or 3D as

$$\mathbf{H}^{\circ}_{2D} = \begin{bmatrix}\left(m_x\nabla\frac{\partial\Phi_x}{\partial x} + m_y\nabla\frac{\partial\Phi_y}{\partial x}\right)^{\top}\\ \left(m_x\nabla\frac{\partial\Phi_x}{\partial y} + m_y\nabla\frac{\partial\Phi_y}{\partial y}\right)^{\top}\end{bmatrix} \tag{8}$$

$$\mathbf{H}^{\circ}_{3D} = \begin{bmatrix}\left(m_x\nabla\frac{\partial\Phi_x}{\partial x} + m_y\nabla\frac{\partial\Phi_y}{\partial x} + m_z\nabla\frac{\partial\Phi_z}{\partial x}\right)^{\top}\\ \left(m_x\nabla\frac{\partial\Phi_x}{\partial y} + m_y\nabla\frac{\partial\Phi_y}{\partial y} + m_z\nabla\frac{\partial\Phi_z}{\partial y}\right)^{\top}\\ \left(m_x\nabla\frac{\partial\Phi_x}{\partial z} + m_y\nabla\frac{\partial\Phi_y}{\partial z} + m_z\nabla\frac{\partial\Phi_z}{\partial z}\right)^{\top}\end{bmatrix}, \tag{9}$$

which allows us to drop the distance vector $\Delta\mathbf{x}$. Note that m_x, m_y and m_z represent the components of the already computed $\mathbf{m}^{\circ}$ vector and the different $\nabla\frac{\partial\Phi_e}{\partial e}, e \in (x,y,z)$ vectors contain the second order derivatives of the flow map, e.g., $\nabla\frac{\partial\Phi_x}{\partial x} = \left(\frac{\partial^2\Phi_x}{\partial x^2}, \frac{\partial^2\Phi_x}{\partial y\partial x}, \frac{\partial^2\Phi_x}{\partial z\partial x}\right)$.

Figure 1 shows the initial location of the extra particles $\mathbf{c}_i$ and $\mathbf{h}_i$ at every time step needed to compute the Jacobian and the Hessian terms in 3D for a central particle at position $\mathbf{c}_0$. As shown in the image, we follow a central finite-difference approach with a separation distance of δx. All the particles, including the extra particles, are advected forward in time with the flow using Runge-Kutta integration obtaining $\hat{\mathbf{c}}_i$ and $\hat{\mathbf{h}}_i$.

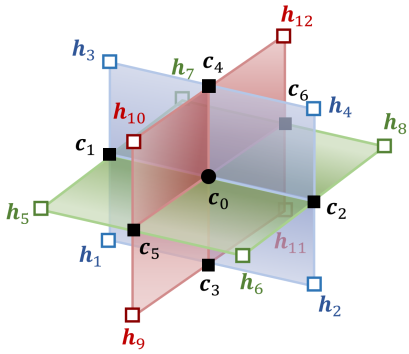


Figure 1: *Extra particles initialization to compute Jacobian and Hessian of the flow map. The particles are advected with the flow to obtain the deformed positions* $\hat{\mathbf{c}}_i$ *and* $\hat{\mathbf{h}}_i$ *at the following time step.*

We show now the discretization for a 3D simulation. Since the positions $\hat{\mathbf{c}}_i$ and $\hat{\mathbf{h}}_i$ correspond to the position of the particles in the new deformed state with respect to the original finite-difference position, we can compute the Jacobian of the flow map as

$$\mathcal{J}^{\circ}_{3D} \approx \frac{1}{\delta x}\begin{bmatrix}(\hat{\mathbf{c}}_2 - \hat{\mathbf{c}}_1)^{\top}\\ (\hat{\mathbf{c}}_4 - \hat{\mathbf{c}}_3)^{\top}\\ (\hat{\mathbf{c}}_6 - \hat{\mathbf{c}}_5)^{\top}\end{bmatrix}^{-1}. \tag{10}$$

The Hessian terms can be discretized in the following way:

$$\left(\frac{\partial^2\Phi_x}{\partial x^2}, \frac{\partial^2\Phi_y}{\partial x^2}, \frac{\partial^2\Phi_z}{\partial x^2}\right) \approx \frac{4}{(\delta x)^2}\left(\hat{\mathbf{c}}_2 - 2\hat{\mathbf{c}}_0 - \hat{\mathbf{c}}_1\right) \tag{11}$$

$$\left(\frac{\partial^2\Phi_x}{\partial y^2}, \frac{\partial^2\Phi_y}{\partial y^2}, \frac{\partial^2\Phi_z}{\partial y^2}\right) \approx \frac{4}{(\delta x)^2}\left(\hat{\mathbf{c}}_4 - 2\hat{\mathbf{c}}_0 - \hat{\mathbf{c}}_3\right) \tag{12}$$

$$\left(\frac{\partial^2\Phi_x}{\partial z^2}, \frac{\partial^2\Phi_y}{\partial z^2}, \frac{\partial^2\Phi_z}{\partial z^2}\right) \approx \frac{4}{(\delta x)^2}\left(\hat{\mathbf{c}}_6 - 2\hat{\mathbf{c}}_0 - \hat{\mathbf{c}}_5\right) \tag{13}$$

$$\left(\frac{\partial^2\Phi_x}{\partial x\partial y}, \frac{\partial^2\Phi_y}{\partial x\partial y}, \frac{\partial^2\Phi_z}{\partial x\partial y}\right) \approx \frac{1}{2\delta x}\left(\hat{\mathbf{h}}_1 - \hat{\mathbf{h}}_2 - \hat{\mathbf{h}}_3 + \hat{\mathbf{h}}_4\right) \tag{14}$$

$$\left(\frac{\partial^2\Phi_x}{\partial x\partial z}, \frac{\partial^2\Phi_y}{\partial x\partial z}, \frac{\partial^2\Phi_z}{\partial x\partial z}\right) \approx \frac{1}{2\delta x}\left(\hat{\mathbf{h}}_5 - \hat{\mathbf{h}}_6 - \hat{\mathbf{h}}_7 + \hat{\mathbf{h}}_8\right) \tag{15}$$

$$\left(\frac{\partial^2\Phi_x}{\partial y\partial z}, \frac{\partial^2\Phi_y}{\partial y\partial z}, \frac{\partial^2\Phi_z}{\partial y\partial z}\right) \approx \frac{1}{2\delta x}\left(\hat{\mathbf{h}}_9 - \hat{\mathbf{h}}_{10} - \hat{\mathbf{h}}_{11} + \hat{\mathbf{h}}_{12}\right) \tag{16}$$

3. Extended Results

In Figure 2 we include the energy plots for the Taylor Vortices example on the main paper.

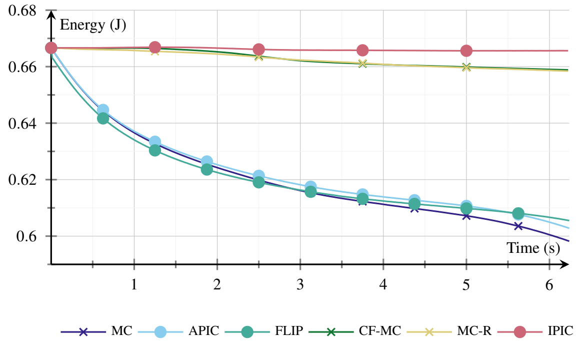


Figure 2: *Energy plots for the Taylor vortices example. IPIC has better energy preservation than the state of the art Covector fluids method.*

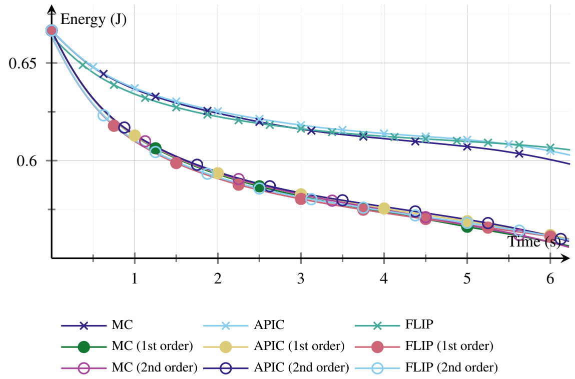


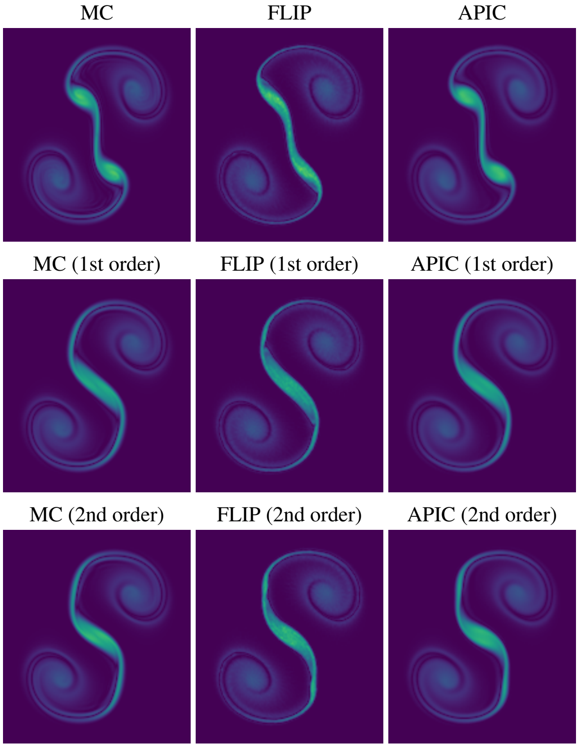
Figure 3: *Energy plots that shows the ablation on 2D Taylor vortices baseline methods*

4. Baseline Comparisons

Throughout the paper, the baseline methods Semi-Lagrangian, MacCormack, PIC, FLIP, and APIC use a first-order projection scheme, performing two consecutive substeps with a half-time step size. For each step, this setting is as expensive as performing the second order scheme that we propose for our method, as it involves two advection and two projection stages. We explored two other reasonable configurations, which under-perform the baseline approach. Figure 3 shows an energy comparison for the Taylor vortices example using the different baseline configurations, and the corresponding vorticity maps are displayed in Figure 4.

APIC (and the other baseline methods) with a *first-order projection scheme and a single step* involves less computation, as it performs only one advection and projection stage. With less numbers of transfers of quantities between grid and particles, this scheme can potentially be beneficial to reduce numerical error. Empirical results show that it is not the case and performing two substeps improves energy preservation.

APIC (and the other baseline methods) using the *second order projection scheme* does not correspond to a second order accuracy upgrade. Note that the proposed scheme is only suitable when combined with the impulse variable — which is divergent by definition — introducing degrees of freedom that can be exploited by the projection operator. More details about the intuition behind this issue can be found in Section 4.5 of [NWRC22]. Our experiments support this theoretical conclusion.


MC FLIP APIC

MC (1st order) FLIP (1st order) APIC (1st order)

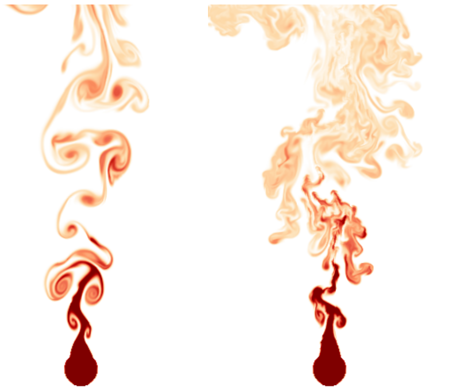
MC (2nd order) FLIP (2nd order) APIC (2nd order)

Figure 4: *2D Taylor vortices with the different baseline configurations at $t = 6$. First row shows the baseline setting used in the main paper, where two first order substeps are performed in one timestep. A single first order step is used in the second row. The third rows shows the results with a second order projection scheme.*

5. Eulerian Stretching

As explored by [FLX*22], it is possible to solve the stretching term in an Eulerian way, independent of the advection step. However, this approach suffers from the splitting error that occurs when solving advection and stretching in a decoupled way. Recent work [DYZ*23] investigated this issue, and found that an inaccurate flow map computation not corresponding to the proper flow map can introduce dissipation.

In our initial attempts, we discretized the impulse stretching in the particle setting, applying a similar idea as Covector Fluids [NWRC22]. We first performed the usual APIC advection using particles and splatted the velocities to the grid. Then, the Jacobian matrices of the flow map on the grid are computed using a Semi-Lagrangian approach, applying them to the grid velocity. However, this solution is not so effective in preserving energy, as there is a clear mismatch between the forward scheme used to advect velocities and the backward scheme used to compute the Jacobian. The result of this approach is shown in Figure 5.



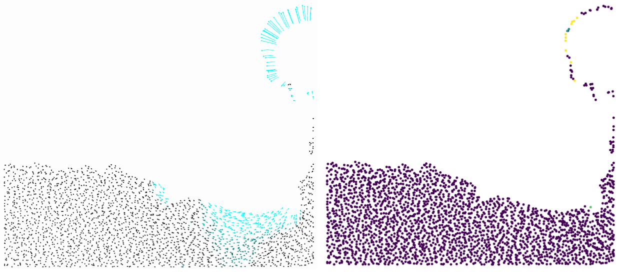
(a) *Grid-based Jacobian* **(b)** *Particle-based Jacobian*

Figure 5: *Performing coupled advection-stretching in the particle setting, which involves computing and applying the Jacobian of the flow map in a per-particle way, significantly reduces the dissipation of the method. Applying the Jacobian on the grid after particle advection, as shown in (a), showcases a more dissipative method than our approach in (b).*

6. Fluid-air Interface Treatment

Particle-In-Cell methods generally extrapolate velocities from fluid cells into air cells to interpolate velocity values for particles close to the free-boundary surface. This solution, however, does not provide precise Jacobian computation at the free-boundary surface for impulse stretching. Furthermore, the gradients computed by APIC can become inaccurate in the absence of a sufficient number of particles at the boundaries.

For these reasons, we revert to FLIP at the interface between air and fluid cells, specifically within a threshold of 2 cells. Although this approach reduces the accuracy of the simulation at the liquid surface, our empirical results show that our method still better preserves the vorticity of the simulation while ensuring method stability. Figure 6 shows an unsuccessful experiments that occurs when FLIP is not used at the fluid-air surface.



(a) *Particle Velocities* **(b)** *Jacobian Determinant*

Figure 6: *Determinant of Jacobian of flow map on the fluid-air interface. (a) visualizes the particle velocity magnitude of the 2-D liquid simulation. Light blue indicates higher velocity, black indicates lower velocity. (b) visualizes the corresponding determinant of the Jacobian of the flow map. Purple indicates the determinant is 1 (correctly computed), yellow indicates being far away from 1 (wrongly computed), and blue indicates close to 1.*

7. Stability and CFL Condition

Our method makes use of the impulse variable to avoid the excessive and uncontrollable dissipation of the former methods, enforcing stability by using a Jacobian-Aware Blending strategy with α and β limiters.

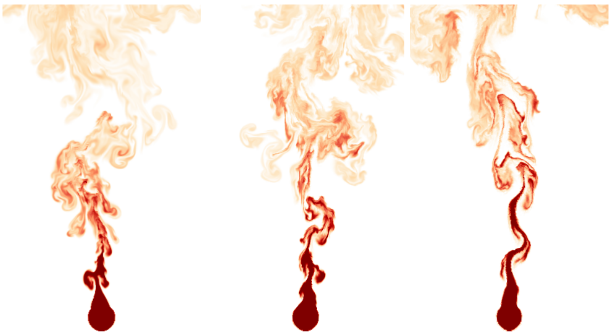
Table 1 shows an analysis of simulation settings of different CFL conditions on the 2D Plume example. In all our simulations, we fix $\Delta x = 1$, and we study how the CFL number changes with the time step size. We compute the CFL condition C as

$$C = \frac{|\mathbf{u}_{\mathbf{max}}|\,\Delta t}{\Delta x},$$

where $|\mathbf{u}_{\mathbf{max}}|$ is the maximum velocity magnitude obtained during the simulation. For those unstable simulations, we indicate the time step where they explode. Impulse-based methods such as Covector Fluids (CF) require using a prohibitively small time step to remain stable. Our blending strategy, with a set of properly-chosen parameters, can ensure stability while obtaining better energy preservation. However, as similarly noticed in [QZG*19], an extremely large time step size can still produce inaccurate results, adding artificial vortex structures that do not correspond to physically-accurate results. Figure 7 shows the visual result of the simulation for a varying set of time step sizes.

Table 1: *CFL conditions and stability.*

Method	Δt	CFL	Stability
CF	0.25	8.6	Unstable at $t = 66$
	0.5	9.6	Unstable at $t = 35$
	1.0	14.8	Unstable at $t = 24$
IPIC ($\alpha = 0.9, \beta = 0.95$)	0.25	15.7	Stable
	0.5	40.7	Unstable at $t = 75$
	1.0	34.2	Unstable at $t = 39$
IPIC ($\alpha = 0.99, \beta = 0.998$)	0.25	7.7	Stable
	0.5	12.5	Stable
	1.0	35.9	Stable
APIC	0.125	3.5	Stable
	0.25	6.7	Stable
	0.5	15.6	Stable



(a) $\Delta t = 0.25$ **(b)** $\Delta t = 0.5$ **(c)** $\Delta t = 1.0$

Figure 7: *Simulation at time* $t = 112$ *using IPIC with* $\alpha = 0.99$ *and* $\beta = 0.998$ *for different time step sizes.*

References

[Cor95] Cortez R.: *Impulse-Based Methods for Fluid Flow*. Tech. rep., 1995. 1

[DYZ*23] Deng Y., Yu H.-X., Zhang D., Wu J., Zhu B.: Fluid simulation on neural flow maps. *ACM Trans. Graph. 42*, 6 (2023). 3

[FLX*22] Feng F., Liu J., Xiong S., Yang S., Zhang Y., Zhu B.: Impulse Fluid Simulation. *IEEE Transactions on Visualization and Computer Graphics* (2022). doi:10.1109/TVCG.2022.3149466. 3

[NWRC22] Nabizadeh M. S., Wang S., Ramamoorthi R., Chern A.: Covector fluids. *ACM Transactions on Graphics 41*, 4 (jul 2022), 1–16. doi:10.1145/3528223.3530120. 3

[QZG*19] Qu Z., Zhang X., Gao M., Jiang C., Chen B.: Efficient and conservative fluids using bidirectional mapping. *ACM Transactions on Graphics 38*, 4 (2019), 1–12. doi:10.1145/3306346.3322945. 4