A Second-Order Advection-Reflection Solver

RAHUL NARAIN, Indian Institute of Technology
JONAS ZEHNDER, Université de Montréal
BERNHARD THOMASZEWSKI, Université de Montréal

Zehnder et al. [2018] recently introduced an advection-reflection method for fluid simulation that dramatically reduces artificial dissipation. We establish a connection between their method and the implicit midpoint time integration scheme, and present a simple modification to obtain an advection-reflection scheme with second-order accuracy in time. We compare with existing alternatives, including a second-order semi-Lagrangian method based on BDF2, and demonstrate the improved energy-preservation properties.

CCS Concepts: • **Computing methodologies → Physical simulation**; *Computer graphics*; *Animation.*

Additional Key Words and Phrases: fluid simulation, advection, reflection, second-order, implicit midpoint

ACM Reference Format:
Rahul Narain, Jonas Zehnder, and Bernhard Thomaszewski. 2019. A Second-Order Advection-Reflection Solver. *Proc. ACM Comput. Graph. Interact. Tech.* 2, 2, Article 16 (July 2019), 14 pages. https://doi.org/10.1145/3340257

1 INTRODUCTION

Visual fluid simulation is the simultaneous pursuit of efficiency, stability, and detail. Large strides toward this goal were made with the introduction of Stam's operator splitting scheme [Stam 1999] that combines semi-Lagrangian advection and pressure projection. While unprecedented in terms of stability and efficiency, the first-order nature of this scheme in both space and time leads to suppression of detail. Subsequent work improved on the spatial and temporal accuracy of the advection step with higher-order schemes [Fedkiw et al. 2001; Kim et al. 2008], temporal error correction [Kim et al. 2005, 2007; Selle et al. 2008], and particle-in-cell methods [Fu et al. 2017; Jiang et al. 2015; Zhu and Bridson 2005]. While alternative representations using vorticity have been proposed [Angelidis and Neyret 2005; Chern et al. 2016; Elcott et al. 2007; Park and Kim 2005; Weißmann and Pinkall 2010], semi-Lagrangian methods remain the dominant approach for smoke animation.

Identifying energy loss due to pressure projection as a main cause of error, Zhang et al. [2015] track and re-inject vorticity to increase the visual richness of animations. The approach by Sato et al. [2018] integrates pressure over multiple time steps, leading to reduced dissipation and, consequently, improved detail preservation. Departing from the common advection-projection paradigm, Zehnder et al. [2018] show that constraint reflection leads to better energy preservation.

Since all of these methods improve on either advection or constraint projection but not both, none of them achieves overall second-order accuracy. By establishing a connection between the

Authors' addresses: Rahul Narain, Indian Institute of Technology, narain@iitd.ac.in; Jonas Zehnder, Université de Montréal, jonas.zehnder@umontreal.ca; Bernhard Thomaszewski, Université de Montréal, bernhard@iro.umontreal.ca.

2577-6193/2019/7-ART16 $15.00
https://doi.org/10.1145/3340257

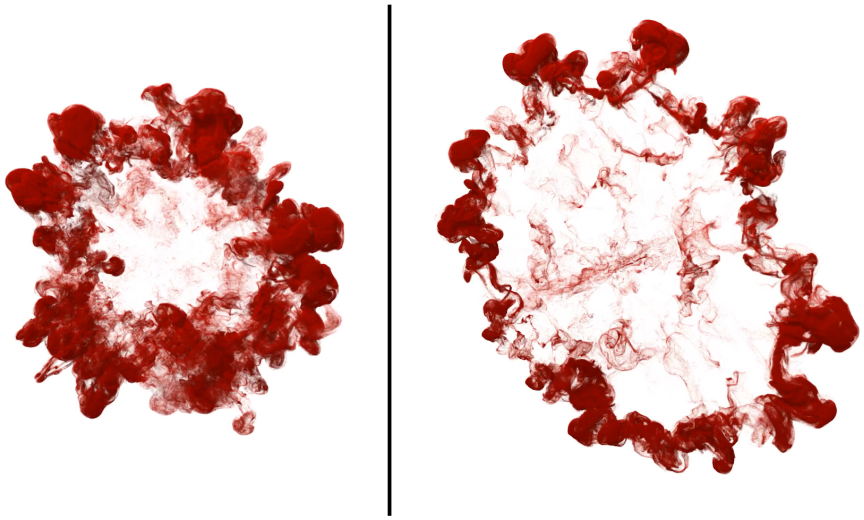

Fig. 1. The result of a collision between two vortex rings. *Left*: BDF2. *Right*: our second-order advection-reflection method. The emergence of the secondary vortical structure around the circumference of the primary vortices phenomenologically matches the behavior observed in real experiments [Lim and Nickels 1992].

advection-reflection scheme of Zehnder et al. [2018] and the implicit midpoint scheme, we derive a simple modification that leads to an overall second-order accurate method. Complementing our formal proof, we furthermore provide numerical evidence for the improved accuracy of our new advection-reflection scheme on a set of 2D and 3D examples. Our comparisons with an alternative second-order accurate method based on BDF2 [Min and Gibou 2006; Xiu and Karniadakis 2001] indicate that the reflective nature of our method is indeed vital for detail-preservation and stability.

2 SECOND-ORDER SEMI-LAGRANGIAN SCHEMES

2.1 Background and Notation

We consider the incompressible Euler equations

$$\rho \frac{\mathrm{D}\mathbf{u}}{\mathrm{D}t} = -\nabla p + \mathbf{f}, \tag{1a}$$

$$\nabla \cdot \mathbf{u} = 0, \tag{1b}$$

and for simplicity assume $\rho = 1$. We use $\mathbf{x}(t)$ to denote a characteristic curve, i.e. the trajectory of a Lagrangian particle satisfying $\dot{\mathbf{x}}(t) = \mathbf{u}(\mathbf{x}(t), t)$. Using the total derivative, we can alternatively express (1) as

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{u}(\mathbf{x}(t), t) = -\nabla p(\mathbf{x}(t)) + \mathbf{f}(\mathbf{x}(t)) \tag{2}$$

subject to the incompressibility constraint $\nabla \cdot \mathbf{u} = 0$.

Consider a time step from time t^0 to $t^1 = t^0 + \Delta t$. For brevity, we will denote $\mathbf{x}(t^0) = \mathbf{x}^0$, $\mathbf{u}(\cdot, t^0) = \mathbf{u}^0(\cdot)$, and similarly for $\mathbf{x}^1$ and $\mathbf{u}^1$. To numerically solve equation (2) over the time step, we must (i) approximate the Lagrangian trajectory $\mathbf{x}(t)$, and (ii) discretize the time derivative $\mathrm{d}/\mathrm{d}t$.

In semi-Lagrangian schemes, the trajectory through the time-varying velocity field $\mathbf{u}$ is approximated by backtracing through a "frozen" velocity field $\mathbf{v}$ instead. Starting from the final position $\mathbf{x}^1$,

we integrate $\dot{\mathbf{x}} = \mathbf{v}(\mathbf{x})$ backward in time to find the approximate initial position $\tilde{\mathbf{x}}^0$. Depending on the choice of $\mathbf{v}$, this is either a first-order or second-order approximation of the true initial position $\mathbf{x}^0$. Then for any scalar or vector field f at time t^0, we can define the advection operator $A[f; \mathbf{v}, \Delta t]$ via

$$A[f; \mathbf{v}, \Delta t](\mathbf{x}^1) = f(\tilde{\mathbf{x}}^0), \tag{3}$$

which represents the field transported along the Lagrangian trajectories to time t^1.

For future use, we also define the pressure projection operator $P[\mathbf{v}]$ which yields the divergence-free component of a vector field $\mathbf{v}$, computed via a Poisson solve

$$P[\mathbf{v}] = \mathbf{v} - \nabla p, \tag{4a}$$

$$\text{where} \quad \nabla^2 p = \nabla \cdot \mathbf{v}. \tag{4b}$$

As discussed by Zehnder et al. [2018], P is an orthogonal projection with respect to the kinetic energy norm and therefore gives rise to an energy-preserving reflection operator $R = 2P - I$.

Different choices for the discretization of the time derivative $\mathrm{d}/\mathrm{d}t$ lead to different semi-Lagrangian schemes. We discuss these choices next.

2.2 Advection-Projection as Backward Euler

The conventional advection-projection splitting scheme [Bridson 2015; Stam 1999] can be obtained by discretizing $\mathrm{d}/\mathrm{d}t$ via backward Euler,

$$\frac{\mathbf{u}^1(\mathbf{x}^1) - \mathbf{u}^0(\mathbf{x}^0)}{\Delta t} = -\nabla p^1(\mathbf{x}^1) + \mathbf{f}^1(\mathbf{x}^1), \tag{5a}$$

$$\nabla \cdot \mathbf{u}^1(\mathbf{x}^1) = 0. \tag{5b}$$

Here, the material point locations $\mathbf{x}^1$ at the end of the time step are given by the grid nodes. The unknown departure points $\mathbf{x}^0$ are approximated to first order by backtracing through $\mathbf{u}^0$ itself. This leads to the scheme

$$\mathbf{u}_a^0 = A[\mathbf{u}^0; \mathbf{u}^0, \Delta t], \tag{6a}$$

$$\mathbf{u}^1 = P[\mathbf{u}_a^0 + \Delta t\, \mathbf{f}^1]. \tag{6b}$$

2.3 BDF2

Before semi-Lagrangian methods were adopted in graphics, they were widely used in meteorology [Staniforth and CÃťtÃľ 1991] with second-order variants being the predominant choice [Robert 1981, 1982]. For incompressible fluids, Xiu and Karniadakis [2001] used BDF2 with spectral element discretization in space, and subsequent work by Min and Gibou [2006] did so with an octree discretization. In both cases, the temporal discretization is of the following form:

$$\frac{\frac{3}{2}\mathbf{u}^1(\mathbf{x}^1) - 2\mathbf{u}^0(\mathbf{x}^0) + \frac{1}{2}\mathbf{u}^{-1}(\mathbf{x}^{-1})}{\Delta t} = -\nabla p^1(\mathbf{x}^1) + \mathbf{f}^1(\mathbf{x}^1), \tag{7a}$$

$$\nabla \cdot \mathbf{u}^1(\mathbf{x}^1) = 0, \tag{7b}$$

where the superscript -1 denotes values at time t^{-1}, the start of the previous time step.

For second-order accuracy, the departure point $\mathbf{x}^{-1}$ is approximated by backtracing $\mathbf{x}^1$ through $\mathbf{u}^0$ for time $2\Delta t$; similarly, $\mathbf{x}^0$ is approximated by backtracing through $\frac{3}{2}\mathbf{u}^0 - \frac{1}{2}\mathbf{u}^{-1}$ for time Δt. This

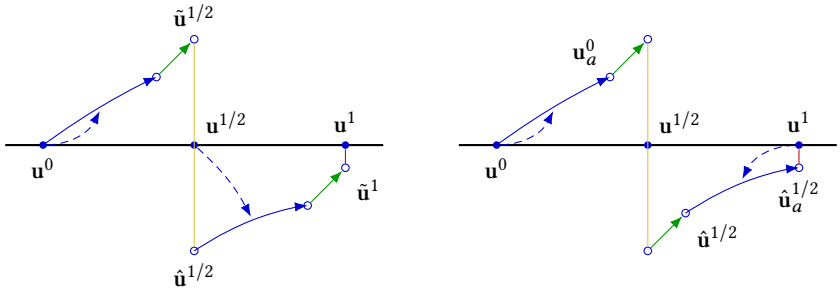

Fig. 2. A geometric interpretation of advection-reflection solvers in the presence of external forces. *Left:* The first-order advection-reflection method performs advection (blue) using the nearest divergence-free velocity field (dashed blue), and applies external forces (green) after advection in both half-steps. *Right:* Our second-order method performs advection using an approximation of $\mathbf{u}^1$ in the second-half-step, and always applies external forces at the mid-step.

yields the method

$$\mathbf{u}_a^{-1} = A[\mathbf{u}^{-1}; \mathbf{u}^0, 2\Delta t], \tag{8a}$$

$$\mathbf{u}_a^0 = A[\mathbf{u}^0; \tfrac{3}{2}\mathbf{u}^0 - \frac{1}{2}\mathbf{u}^{-1}, \Delta t], \tag{8b}$$

$$\mathbf{u}^1 = P[\tfrac{4}{3}\mathbf{u}_a^0 - \tfrac{1}{3}\mathbf{u}_a^{-1} + \tfrac{2}{3}\Delta t\, \mathbf{f}^1]. \tag{8c}$$

2.4 Advection-Reflection as Implicit Midpoint

The advection-reflection method introduced by Zehnder et al. [2018] implements the following update rule:

$$\tilde{\mathbf{u}}^{1/2} = A[\mathbf{u}^0; \mathbf{u}^0, \tfrac{1}{2}\Delta t] + \tfrac{1}{2}\Delta t\, \mathbf{f}^{1/2}, \tag{9a}$$

$$\mathbf{u}^{1/2} = P[\tilde{\mathbf{u}}^{1/2}], \tag{9b}$$

$$\hat{\mathbf{u}}^{1/2} = 2\mathbf{u}^{1/2} - \tilde{\mathbf{u}}^{1/2}, \tag{9c}$$

$$\tilde{\mathbf{u}}^1 = A[\hat{\mathbf{u}}^{1/2}; \mathbf{u}^{1/2}, \tfrac{1}{2}\Delta t] + \tfrac{1}{2}\Delta t\, \mathbf{f}^{1/2}, \tag{9d}$$

$$\mathbf{u}^1 = P[\tilde{\mathbf{u}}^1]. \tag{9e}$$

The steps of this method are illustrated schematically in Fig. 2 (left). While this scheme preserves energy better than conventional advection-projection methods, it inherits their first-order accuracy.

To achieve higher-order accuracy in the advection-reflection scheme, we start by drawing an analogy to the second-order accurate implicit midpoint method. To this end we discretize the total derivative $\mathrm{d}/\mathrm{d}t$ by evaluating all velocities, positions, and forces at mid-step. Accordingly, we apply

a first constraint projection at the middle of the time interval and a final projection at the end:

$$\frac{\mathbf{u}^{1/2}(\mathbf{x}^{1/2}) - \mathbf{u}^0(\mathbf{x}^0)}{\frac{1}{2}\Delta t} = -\nabla p^{1/2}(\mathbf{x}^{1/2}) + \mathbf{f}^{1/2}(\mathbf{x}^{1/2}), \tag{10a}$$

$$\nabla \cdot \mathbf{u}^{1/2}(\mathbf{x}^{1/2}) = 0, \tag{10b}$$

$$\frac{\tilde{\mathbf{u}}^1(\mathbf{x}^1) - \mathbf{u}^{1/2}(\mathbf{x}^{1/2})}{\frac{1}{2}\Delta t} = -\nabla p^{1/2}(\mathbf{x}^{1/2}) + \mathbf{f}^{1/2}(\mathbf{x}^{1/2}), \tag{10c}$$

$$\mathbf{u}^1(\mathbf{x}^1) - \tilde{\mathbf{u}}^1(\mathbf{x}^1) = \nabla\lambda(\mathbf{x}^1), \tag{10d}$$

$$\nabla \cdot \mathbf{u}^1(\mathbf{x}^1) = 0. \tag{10e}$$

Achieving second-order accuracy requires sufficiently accurate approximations of $\mathbf{x}^0$ and $\mathbf{x}^{1/2}$. While semi-Lagrangian advection typically yields only first-order accuracy, accuracy can be elevated by composing forward and backward approximations of the Lagrangian trajectories, i.e. advecting through $\mathbf{u}^0$ and $\mathbf{u}^1$ in the two half-steps. As we show in Sec. 3.1, this causes the leading-order errors to cancel out. In fact, it is sufficient to use a second-order accurate approximation of $\mathbf{u}^1$, namely $2\mathbf{u}^{1/2} - \mathbf{u}^0$. This yields our proposed method:

$$\mathbf{u}_a^0 = A[\mathbf{u}^0; \mathbf{u}^0, \tfrac{1}{2}\Delta t], \tag{11a}$$

$$\mathbf{u}^{1/2} = P[\mathbf{u}_a^0 + \tfrac{1}{2}\Delta t\, \mathbf{f}^{1/2}], \tag{11b}$$

$$\hat{\mathbf{u}}^{1/2} = 2\mathbf{u}^{1/2} - \mathbf{u}_a^0, \tag{11c}$$

$$\hat{\mathbf{u}}_a^{1/2} = A[\hat{\mathbf{u}}^{1/2}; 2\mathbf{u}^{1/2} - \mathbf{u}^0, \tfrac{1}{2}\Delta t], \tag{11d}$$

$$\mathbf{u}^1 = P[\hat{\mathbf{u}}_a^{1/2}]. \tag{11e}$$

It should be noted that (11c) is equivalent to $\hat{\mathbf{u}}^{1/2} = R[\mathbf{u}_a^0] + \Delta t\, P[\mathbf{f}^{1/2}]$. Moreover, the mid-step velocity $\mathbf{u}^{1/2}$ is the same as that in the original advection-reflection method—the only changes are the velocity used for advection in (d) and the application of external forces, which are now correctly evaluated at mid-step. This method is illustrated schematically in Fig. 2 (right), making apparent its more symmetric nature.

3 ANALYSIS

3.1 Accuracy and Stability

In Appendix A we show that the modified advection-reflection method has second-order temporal accuracy. The key steps of our accuracy analysis are summarized below.

First, implicit midpoint with constraint projections is by itself second-order accurate, so if we could directly solve the equations (10) using the true characteristics, we would have a second-order accurate solution for $\mathbf{u}^1$. Let the mid-step and final velocities computed by this hypothetical algorithm be $\mathbf{v}^{1/2}$ and $\mathbf{v}^1$. Then the error in $\mathbf{v}^1$ is $O(\Delta t^3)$.

Second, we estimate the error caused by backtracing through the frozen velocity fields $\mathbf{u}^0$ and $2\mathbf{u}^{1/2} - \mathbf{u}^0$. These are shown to be $\frac{1}{8}\Delta t^2 \nabla\mathbf{u} \cdot \dot{\mathbf{u}} + O(\Delta t^3)$ and $-\frac{1}{8}\Delta t^2 \nabla\mathbf{u} \cdot \dot{\mathbf{u}} + O(\Delta t^3)$, respectively. Thus, the $O(\Delta t^2)$ terms in the total error cancel out, and the local error of the algorithm is $O(\Delta t^3)$. Note that for this bound to hold with discretized backtracing, the backtracing itself must also be second-order accurate; we use explicit midpoint in our implementation.

For stability, the proof in the absence of external forces is the same as Zehnder et al. [2018]: the final projection reduces energy, all other operations conserve or decrease energy. In the presence of external forces, we experimentally observed that the total energy is generally not monotonic for our

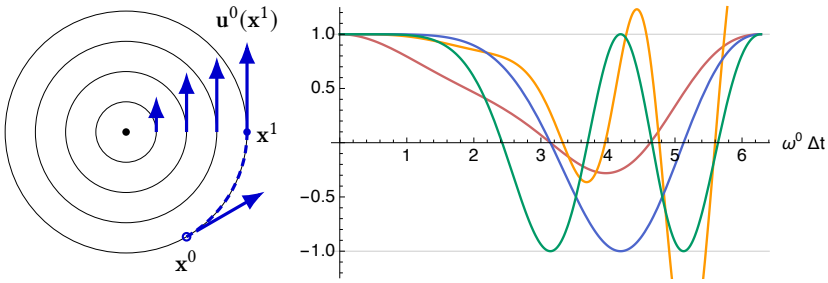

Fig. 3. Velocity amplification factor in circular flow. We take two time steps of length $\Delta t/2$ for advection-projection and BDF2, versus one time step of length Δt for advection-reflection solvers. *Red*: advection-projection. *Orange*: BDF2. *Blue*: advection-reflection. *Green*: second-order advection-reflection.

method. However, the same limitations apply for BDF2 and first-order advection-reflection. Indeed, even energy-preserving fluid simulation methods [Cui et al. 2018; De Witt et al. 2012; Mullen et al. 2009] have not been proven to guarantee stability under conservative external forces. Nevertheless, our experiments with (conservative) buoyancy forces indicate that the deviation from constant energy is smaller for our second-order version than for its competitors; see Fig. 5 (left).

3.2 Circular Flow

To better understand the behavior of these schemes, we consider a scenario that admits a closed-form solution. Let $\mathbf{u}$ be a circularly symmetric flow around the origin with radially varying angular velocity $\omega(r)$, i.e. $\mathbf{u}(r,\theta) = r\omega(r)\mathbf{e}_\theta$ in polar coordinates. This is a steady state, so $\mathbf{u}$ should remain constant over time. We assume that exact backtracing is performed, in which case $\mathbf{x}^0$ is simply $\mathbf{x}^1$ rotated by $-\omega(\|\mathbf{x}^1\|)\Delta t$, so the velocity after advection is independent of ω at other r (Fig. 3, left). Projection simply annihilates the radial component of the velocity, and the flow remains circularly symmetric, $\mathbf{u}^1(r,\theta) = r\omega^1(r)\mathbf{e}_\theta$.

In Fig. 3 (right) we plot the velocity amplification factor, ω^1/ω^0, as a function of $\omega^0\Delta t$. This should equal 1 in the ideal case; $|\omega^1/\omega^0| < 1$ indicates numerical dissipation. Since the advection-reflection methods require two pressure solves per time step, for a fairer comparison we ran advection-projection and BDF2 with two time steps of length $\Delta t/2$. The analysis of BDF2 is complicated by the fact that the first step depends on the velocity at an earlier time; since the true solution is steady, we set this velocity to ω^0 as well.

It can be observed that all second-order accurate methods show very low dissipation when $\omega^0\Delta t < 1.5$. However, BDF2 has $|\omega^1/\omega^0| > 1$ for some $\omega^0\Delta t > 4$, showing that it can amplify high-vorticity flows. This amplification is a potential source of noise, explaining the artifacts that we observed, e.g., in the vortex ring collision example (see accompanying video).

Fig. 3 also reveals that while both advection-reflection methods are very close to energy-preserving for small vorticities, for large ones the second-order version begins to lose energy somewhat earlier. For example, the point where $\omega^1/\omega^0 = 0.9$ occurs at $\omega^0\Delta t \approx 1.99$ for the first-order method, but at $\omega^0\Delta t \approx 1.56$ for the second-order method. Therefore, if the only concern is to preserve as much *kinetic* energy as possible in highly turbulent animations, the first-order method may be preferable. However, as we show in the following section, it exhibits less desirable behavior in the presence of *potential* energy from forces like buoyancy.

4 RESULTS

We validate the behavior of our second-order advection-reflection method in the context of practically-relevant fluid animations by applying to a set of 2D and 3D examples. As in Section 3.2, for all comparisons we run advection-projection and BDF2 with half the time step used for the advection-reflection schemes to obtain comparable computation time. In practice, this choice constitutes a slight advantage for BDF2, since it has an additional advection operation, and a slight disadvantage for advection-reflection, since the final pre-projection velocity ($\tilde{\mathbf{u}}^1$ or $\hat{\mathbf{u}}_a^{1/2}$) is usually close to divergence-free so the second projection is significantly cheaper.

We start with a simple 2D example with nontrivial dynamics, adopted from Straka et al. [1993], albeit without viscosity. A negatively buoyant bubble of cold smoke with initial temperature profile

$$T(x,y) = \begin{cases} -\frac{1}{2}(\cos(\pi L(x,y)) + 1) & \text{if } L(x,y) < 1, \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

where $L(x,y) = (x/4)^2 + ((y-3)/2)^2$, sinks under gravity and spreads along the bottom of the domain. Fig. 4 shows results with different time step sizes. The results for the second-order reflection scheme are less sensitive to time step size than other methods, and less noisy at large time steps than BDF2. Furthermore, as shown in Fig. 5 (left), we observe that the energy behavior of BDF2 varies much more with changes in step size: at large time steps, it loses more energy than second-order advection-reflection, while at small time steps, it can even gain energy. Existing advection-projection and advection-reflection methods are also seen to gain energy early on in the simulation; since they do not apply external forces at the mid-step, the energy is not transferred correctly beween potential and kinetic modes as the bubble first sinks.

We further validate the temporal accuracy of our method by computing the velocity field at a fixed time $t = 32$ using different time steps $\Delta t = 32, 16, 8, 4, 2$. We use the solution computed with BDF2 and a time step of $\Delta t = 1/2$ as the reference. The L^2 difference to the reference solution is plotted in Fig. 5 (right). Our results are consistent with the theoretical order of accuracy: advection-projection and Zehnder et al. [2018]'s advection-reflection are first-order, while BDF2 and our modified advection-reflection are second-order.

In a second 2D example, we apply the second-order advection-reflection solver to the well-known Karman vortex street example and compare its behavior to BDF2. While both methods produce quasi-regular shedding, BDF2 leads to smearing and stretching of the vortices (Fig. 6, bottom). Our method produces clean and well-separated vortices (Fig. 6, bottom), underlining the fact that numerical accuracy alone does not imply structure preservation.

To study the behavior of our method for 3D fluid animations with complex turbulent behavior, we simulate the collision between two vortex rings—an example that exhibits both turbulence and structure. This scenario has been studied by Lim and Nickels [1992] and recordings of the real-world experiments are available online; see, e.g., Sandlin [2018]. In our simulation, we added a small amount of curl noise to the initial velocities to encourage symmetry breaking. As can be seen in the accompanying video, upon collision, the two vortex rings initially spread in the plane normal to their relative velocity followed by the formation of secondary vortices. It can be seen from the video that BDF2 produces an excessive amount of turbulent detail, but it lacks the structure expected from the real-world experiment. By contrast, our method produces the secondary vortical structure around the circumference of the primary vortex rings; see also Fig. 1.

We also apply our method to the standard example of a smoke plume rising due to buoyancy (Fig. 7), and the same scenario in the presence of an obstacle (Fig. 8). Both advection-reflection methods produce qualitatively similar results, showing large-scale vortical structures as well as fine-scale turbulence. It can be seen in the supplementary video that the vortical structures tend

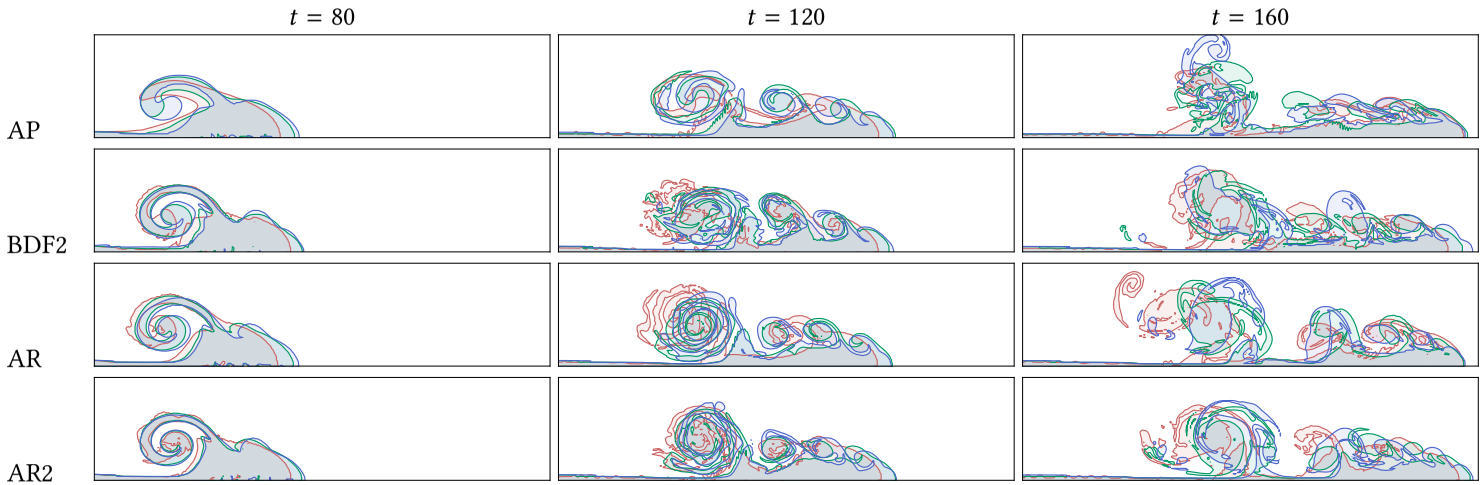

Fig. 4. Comparison of time step dependence of different methods on the cold bubble scenario. Contours at temperature $T = -0.1$ are drawn for three different time step sizes: $\Delta t = 4$ in red, $\Delta t = 2$ in green, and $\Delta t = 1$ in blue. The second-order advection-reflection method shows the most consistent behavior across different time step sizes, as can be seen in the locations of the trailing vortex and the leading front.

Proc. ACM Comput. Graph. Interact. Tech., Vol. 2, No. 2, Article 16. Publication date: July 2019.

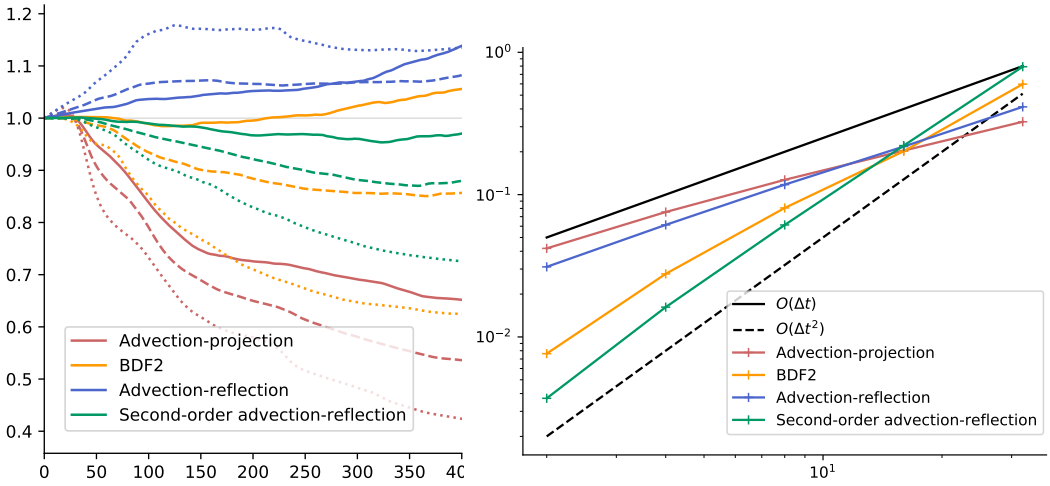

Fig. 5. *Left*: Total energy of different solvers on the cold bubble scenario for different time step sizes, normalized by the initial energy. Dotted, dashed and solid lines represent $\Delta t = 4$, $\Delta t = 2$, and $\Delta t = 1$ respectively. *Right*: Convergence of different solvers as a function of time step.

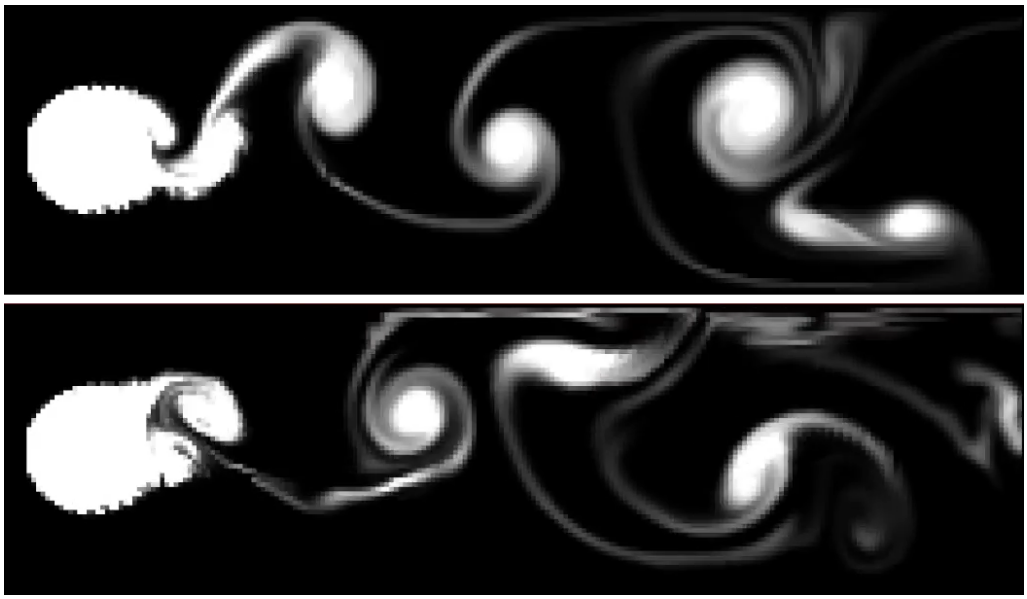

Fig. 6. Karman vortex street example. *Bottom*: BDF2 leads to stretched and smeared-out vortices. *Top*: our second-order advection-reflection solver produces clean and well-separated vortices.

to be more coherent over time for the second-order variant. Nevertheless, for these examples BDF2 gives arguably the most visually compelling results, since the additional noise leads to the production of noticeably stronger turbulence.

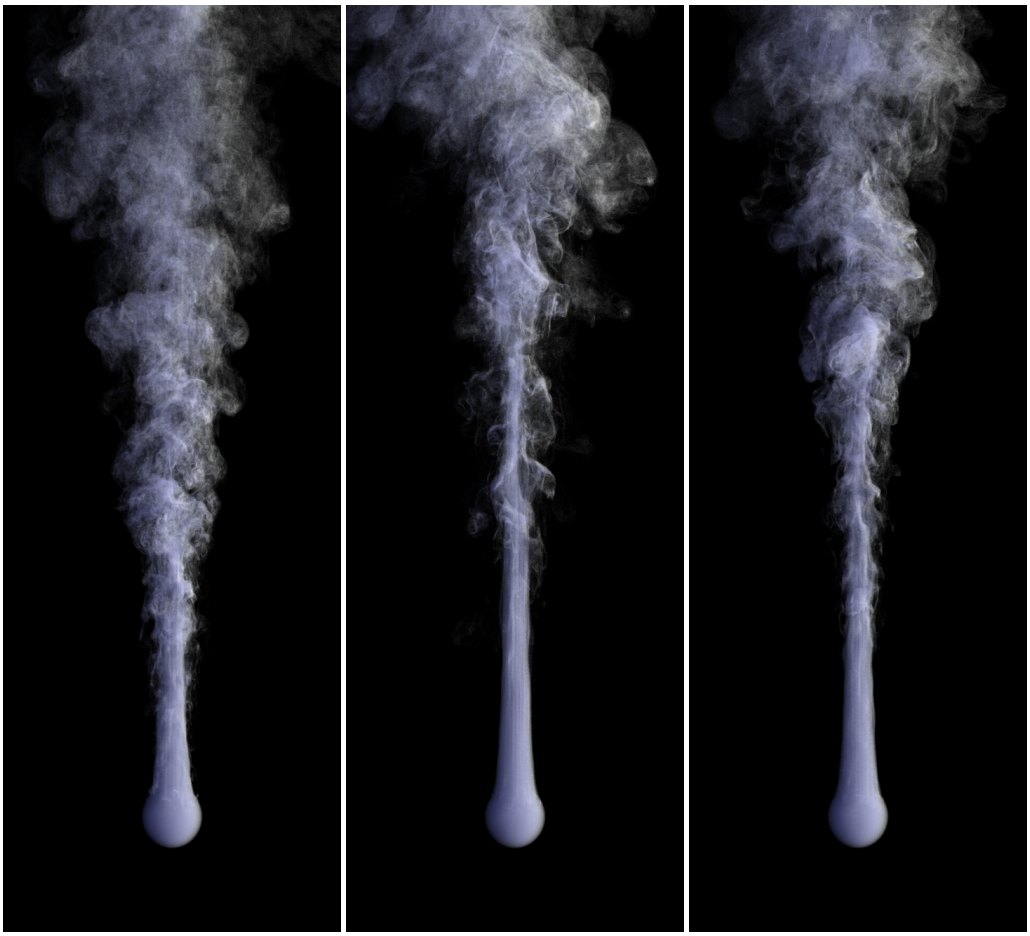

Fig. 7. A rising plume of hot smoke. *Left*: BDF2. *Middle*: Advection-reflection. *Right*: Second-order advection-reflection.

5 CONCLUSION

We presented a simple modification of the first-order advection-reflection method by Zehnder et al. [2018] that achieves second-order accuracy in time. The resulting method is shown to exhibit better energy behavior and structure preservation than existing methods, as well as less sensitivity to time step size. Visually, the results compare favourably to those of the first-order method in all cases, and to those of BDF2 in most. As such, we argue that there is little reason not to use our second-order scheme in place of first-order advection-reflection, as it comes at virtually no extra cost.

ACKNOWLEDGMENTS

We thank Robert Bridson for illuminating discussions, and the anonymous reviewers for their constructive and thoughtful feedback. RN is supported by a Pankaj Gupta Young Faculty Fellowship and a gift from Adobe Inc. BT is supported by the NSERC Discovery Grants and Discovery Accelerator Awards program. JZ is supported by a Google Excellence scholarship.

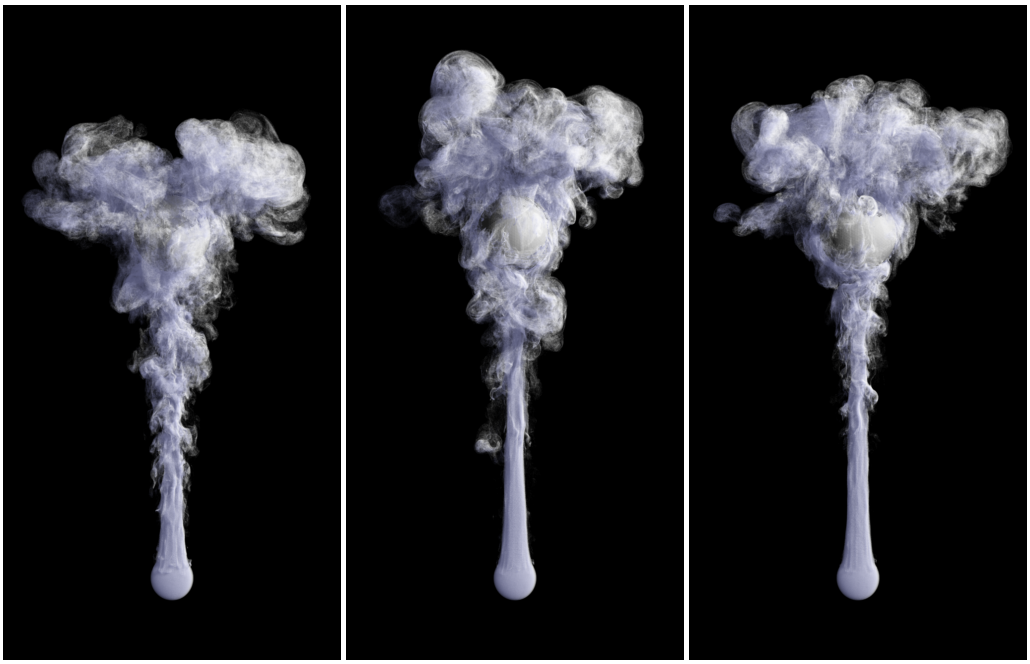

Fig. 8. A smoke plume collides with an obstacle. *Left*: BDF2. *Middle*: Advection-reflection. *Right*: Second-order advection-reflection.

REFERENCES

Alexis Angelidis and Fabrice Neyret. 2005. Simulation of Smoke Based on Vortex Filament Primitives. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA '05)*. 87–96. https://doi.org/10.1145/1073368.1073380

Uri M. Ascher and Linda R. Petzold. 1998. *Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations*. SIAM.

Robert Bridson. 2015. *Fluid Simulation for Computer Graphics* (2nd ed.). CRC Press.

Albert Chern, Felix Knöppel, Ulrich Pinkall, Peter Schröder, and Steffen Weißmann. 2016. Schrödinger's Smoke. *ACM Trans. Graph.* 35, 4 (July 2016), 77:1–77:13. https://doi.org/10.1145/2897824.2925868

Qiaodong Cui, Pradeep Sen, and Theodore Kim. 2018. Scalable Laplacian Eigenfluids. *ACM Trans. Graph.* 37, 4, Article 87 (July 2018), 12 pages. https://doi.org/10.1145/3197517.3201352

Tyler De Witt, Christian Lessig, and Eugene Fiume. 2012. Fluid Simulation Using Laplacian Eigenfunctions. *ACM Trans. Graph.* 31, 1, Article 10 (Feb. 2012), 11 pages. https://doi.org/10.1145/2077341.2077351

Sharif Elcott, Yiying Tong, Eva Kanso, Peter Schröder, and Mathieu Desbrun. 2007. Stable, Circulation-preserving, Simplicial Fluids. *ACM Trans. Graph.* 26, 1 (Jan. 2007). https://doi.org/10.1145/1189762.1189766

Ronald Fedkiw, Jos Stam, and Henrik Wann Jensen. 2001. Visual Simulation of Smoke. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '01)*. 15–22. https://doi.org/10.1145/383259.383260

Chuyuan Fu, Qi Guo, Theodore Gast, Chenfanfu Jiang, and Joseph Teran. 2017. A Polynomial Particle-in-cell Method. *ACM Trans. Graph.* 36, 6, Article 222 (Nov. 2017), 12 pages. https://doi.org/10.1145/3130800.3130878

Chenfanfu Jiang, Craig Schroeder, Andrew Selle, Joseph Teran, and Alexey Stomakhin. 2015. The affine particle-in-cell method. *ACM Transactions on Graphics (TOG)* 34, 4 (July 2015), 51:1–51:10. https://doi.org/10.1145/2766996

ByungMoon Kim, Yingjie Liu, Ignacio Llamas, and Jarek Rossignac. 2005. FlowFixer: Using BFECC for Fluid Simulation. In *Proceedings of the First Eurographics Conference on Natural Phenomena (NPH'05)*. 51–56. https://doi.org/10.2312/NPH/NPH05/051-056

ByungMoon Kim, Yingjie Liu, Ignacio Llamas, and Jarek Rossignac. 2007. Advections with Significantly Reduced Dissipation and Diffusion. *IEEE Transactions on Visualization and Computer Graphics* 13, 1 (Jan. 2007), 135–144. https://doi.org/10.

1109/TVCG.2007.3

Doyub Kim, Oh-young Song, and Hyeong-Seok Ko. 2008. A Semi-Lagrangian CIP Fluid Solver without Dimensional Splitting. *Computer Graphics Forum* 27, 2 (April 2008), 467–475. https://doi.org/10.1111/j.1467-8659.2008.01144.x

T. T. Lim and T. B. Nickels. 1992. Instability and reconnection in the head-on collision of two vortex rings. *Nature* 357 (1992). Issue 6375.

Chohong Min and FrÃĺdÃĺric Gibou. 2006. A second order accurate projection method for the incompressible NavierâĂŞStokes equations on non-graded adaptive grids. *J. Comput. Phys.* 219, 2 (2006), 912 – 929. https://doi.org/10.1016/j.jcp.2006.07.019

Patrick Mullen, Keenan Crane, Dmitry Pavlov, Yiying Tong, and Mathieu Desbrun. 2009. Energy-preserving Integrators for Fluid Animation. *ACM Trans. Graph.* 28, 3, Article 38 (July 2009), 8 pages. https://doi.org/10.1145/1531326.1531344

Sang Il Park and Myoung Jun Kim. 2005. Vortex Fluid for Gaseous Phenomena. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA '05)*. 261–270. https://doi.org/10.1145/1073368.1073406

André Robert. 1981. A stable numerical integration scheme for the primitive meteorological equations. *Atmosphere-Ocean* 19, 1 (1981), 35–46. https://doi.org/10.1080/07055900.1981.9649098 arXiv:https://doi.org/10.1080/07055900.1981.9649098

André Robert. 1982. A Semi-Lagrangian and Semi-Implicit Numerical Integration Scheme for the Primitive Meteorological Equations. *Journal of the Meteorological Society of Japan. Ser. II* 60, 1 (1982), 319–325. https://doi.org/10.2151/jmsj1965.60.1_319

Destin Sandlin. 2018. Smarter Every Day: Two Vortex Rings Colliding in Slow Motion. https://www.youtube.com/watch?v=EVbdbVhzcM4. (2018).

Takahiro Sato, Christopher Batty, Takeo Igarashi, and Ryoichi Ando. 2018. Spatially adaptive long-term semi-Lagrangian method for accurate velocity advection. *Computational Visual Media* 4, 3 (01 Sep 2018), 223–230. https://doi.org/10.1007/s41095-018-0117-9

Andrew Selle, Ronald Fedkiw, Byungmoon Kim, Yingjie Liu, and Jarek Rossignac. 2008. An Unconditionally Stable MacCormack Method. *J. Sci. Comput.* 35, 2-3 (June 2008), 350–371. https://doi.org/10.1007/s10915-007-9166-4

Jos Stam. 1999. Stable Fluids. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '99)*. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 121–128. https://doi.org/10.1145/311535.311548

Andrew Staniforth and Jean CÃťtÃĺ. 1991. Semi-Lagrangian Integration Schemes for Atmospheric ModelsâĂŤA Review. *Monthly Weather Review* 119, 9 (1991), 2206–2223. https://doi.org/10.1175/1520-0493(1991)119<2206:SLISFA>2.0.CO;2

J. M. Straka, Robert B. Wilhelmson, Louis J. Wicker, John R. Anderson, and Kelvin K. Droegemeier. 1993. Numerical solutions of a non-linear density current: A benchmark solution and comparisons. *International Journal for Numerical Methods in Fluids* 17, 1 (1993), 1–22. https://doi.org/10.1002/fld.1650170103 arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/fld.1650170103

Steffen Weißmann and Ulrich Pinkall. 2010. Filament-based Smoke with Vortex Shedding and Variational Reconnection. *ACM Trans. Graph.* 29, 4 (July 2010), 115:1–115:12. https://doi.org/10.1145/1778765.1778852

Dongbin Xiu and George Em Karniadakis. 2001. A Semi-Lagrangian High-Order Method for Navier–Stokes Equations. *J. Comput. Phys.* 172, 2 (2001), 658 – 684. https://doi.org/10.1006/jcph.2001.6847

Jonas Zehnder, Rahul Narain, and Bernhard Thomaszewski. 2018. An Advection-reflection Solver for Detail-preserving Fluid Simulation. *ACM Trans. Graph.* 37, 4, Article 85 (July 2018), 8 pages. https://doi.org/10.1145/3197517.3201324

Xinxin Zhang, Robert Bridson, and Chen Greif. 2015. Restoring the Missing Vorticity in Advection-projection Fluid Solvers. *ACM Trans. Graph.* 34, 4 (July 2015), 52:1–52:8. https://doi.org/10.1145/2766982

Yongning Zhu and Robert Bridson. 2005. Animating Sand as a Fluid. 24 (July 2005), 965–972. https://doi.org/10.1145/1073204.1073298

A ACCURACY OF SECOND-ORDER ADVECTION-REFLECTION

Projected implicit midpoint is a projected Gauss-Legendre method and hence second-order accurate in time [Ascher and Petzold 1998, Ch. 10.1.5]. Thus, if we could directly solve the equations (10) using the true characteristics, we would have a second-order accurate solution for $\mathbf{u}^1$. To distinguish

this hypothetical scheme from the practical algorithm (11), we write it using $\mathbf{v}$ instead of $\mathbf{u}$:

$$\mathbf{v}_a^0(\mathbf{x}^{1/2}) = \mathbf{u}^0(\mathbf{x}^0), \tag{13a}$$
$$\mathbf{v}^{1/2} = P[\mathbf{v}_a^0 + \tfrac{1}{2}\Delta t\, \mathbf{f}^{1/2}], \tag{13b}$$
$$\hat{\mathbf{v}}^{1/2} = R[\mathbf{v}_a^0] + \Delta t\, P[\mathbf{f}^{1/2}], \tag{13c}$$
$$\hat{\mathbf{v}}_a^{1/2}(\mathbf{x}^1) = \hat{\mathbf{v}}^{1/2}(\mathbf{x}^{1/2}), \tag{13d}$$
$$\mathbf{v}^1 = P[\hat{\mathbf{v}}_a^{1/2}]. \tag{13e}$$

Unfortunately, given $\mathbf{x}^{1/2}$ we do not have $\mathbf{x}^0$, and given $\mathbf{x}^1$ we do not have either, because the characteristic curves of the solution $\mathbf{u}$ are not known a priori. Therefore it is also necessary to analyze the error introduced by semi-Lagrangian advection through the constant velocity fields $\mathbf{u}^0$ and $\tilde{\mathbf{u}}^1$.

Consider the first half-step. We would like to find the departure point $\mathbf{x}^0 = \mathbf{x}(t^0)$ by solving the ODE $\dot{\mathbf{x}}(t) = \mathbf{u}(\mathbf{x}(t), t)$ with boundary condition $\mathbf{x}(t^{1/2}) = \mathbf{x}^{1/2}$. Semi-Lagrangian advection instead solves $\dot{\tilde{\mathbf{x}}}(t) = \mathbf{u}^0(\tilde{\mathbf{x}}(t)) = \mathbf{u}(\tilde{\mathbf{x}}(t), t^0)$ with $\tilde{\mathbf{x}}(t^{1/2}) = \mathbf{x}^{1/2}$, yielding an approximation $\tilde{\mathbf{x}}^0$. A Taylor expansion about $(\mathbf{x}^{1/2}, t^{1/2})$ shows that the exact and computed departure points are

$$\mathbf{x}^0 = \mathbf{x}^{1/2} - \tfrac{1}{2}\Delta t\, \mathbf{u}^{1/2} + \tfrac{1}{8}\Delta t^2(\nabla \mathbf{u}^{1/2} \cdot \mathbf{u}^{1/2} + \dot{\mathbf{u}}) + O(\Delta t^3), \tag{14}$$
$$\tilde{\mathbf{x}}^0 = \mathbf{x}^{1/2} - \tfrac{1}{2}\Delta t\, \mathbf{u}^0 + \tfrac{1}{8}\Delta t^2 \nabla \mathbf{u}^0 \cdot \mathbf{u}^0 + O(\Delta t^3), \tag{15}$$

as long as $\tilde{\mathbf{x}}^0$ is computed with at least second-order accurate backtracing. The difference between the two is

$$\begin{aligned}\tilde{\mathbf{x}}^0 - \mathbf{x}^0 &= \tfrac{1}{2}\Delta t(\mathbf{u}^{1/2} - \mathbf{u}^0) - \tfrac{1}{8}\Delta t^2 \dot{\mathbf{u}} + O(\Delta t^3) \\ &= \tfrac{1}{8}\Delta t^2 \dot{\mathbf{u}} + O(\Delta t^3),\end{aligned} \tag{16}$$

since the difference between $\Delta t^2 \nabla \mathbf{u}^{1/2} \cdot \mathbf{u}^{1/2}$ and $\Delta t^2 \nabla \mathbf{u}^0 \cdot \mathbf{u}^0$ can be absorbed into the $O(\Delta t^3)$ term. Finally, having estimated $\tilde{\mathbf{x}}^0 - \mathbf{x}^0$, we can estimate the difference in the advected velocity fields $\mathbf{u}_a^0(\mathbf{x}^{1/2}) = \mathbf{u}^0(\tilde{\mathbf{x}}^0)$ and $\mathbf{v}_a^0(\mathbf{x}^{1/2}) = \mathbf{u}^0(\mathbf{x}^0)$. Taking a Taylor expansion of $\mathbf{u}^0$ about $\mathbf{x}^0$, we find that

$$\begin{aligned}\mathbf{u}_a^0(\mathbf{x}^{1/2}) - \mathbf{v}_a^0(\mathbf{x}^{1/2}) &= \mathbf{u}^0(\tilde{\mathbf{x}}^0) - \mathbf{u}^0(\mathbf{x}^0) \\ &= \Delta t^2 \underbrace{(\tfrac{1}{8}\nabla \mathbf{u}^0 \cdot \dot{\mathbf{u}})}_{\mathbf{e}^0} + O(\Delta t^3).\end{aligned} \tag{17}$$

We then immediately obtain

$$\mathbf{u}^{1/2} - \mathbf{v}^{1/2} = \Delta t^2 P[\mathbf{e}^0] + O(\Delta t^3), \tag{18}$$
$$\hat{\mathbf{u}}^{1/2} - \hat{\mathbf{v}}^{1/2} = \Delta t^2 R[\mathbf{e}^0] + O(\Delta t^3). \tag{19}$$

In the second half-step of our advection-reflection method, we perform advection using the extrapolated velocity field $\tilde{\mathbf{u}}^1 = 2\mathbf{u}^{1/2} - \mathbf{u}^0$. Here $\mathbf{u}^{1/2}$ has been obtained by a stable fluids half-step, which is first-order accurate, so $\mathbf{u}^{1/2} - \mathbf{u}(\cdot, t^{1/2}) = O(\Delta t^2)$. It follows that $\tilde{\mathbf{u}}^1 - \mathbf{u}(\cdot, t^1) = O(\Delta t^2)$. A similar analysis as before then gives

$$\tilde{\mathbf{x}}^{1/2} - \mathbf{x}^{1/2} = -\tfrac{1}{8}\Delta t^2 \dot{\mathbf{u}} + O(\Delta t^3). \tag{20}$$

Finally, we can write the error after the second half-step, $\hat{\mathbf{u}}_a^{1/2}(\mathbf{x}^1) - \hat{\mathbf{v}}_a^{1/2}(\mathbf{x}^1)$, as the sum of two terms:

$$\begin{aligned}
&\hat{\mathbf{u}}_a^{1/2}(\mathbf{x}^1) - \hat{\mathbf{v}}_a^{1/2}(\mathbf{x}^1) \\
&= \hat{\mathbf{u}}^{1/2}(\tilde{\mathbf{x}}^{1/2}) - \hat{\mathbf{v}}^{1/2}(\mathbf{x}^{1/2}) && (21) \\
&= \left(\hat{\mathbf{u}}^{1/2}(\tilde{\mathbf{x}}^{1/2}) - \hat{\mathbf{u}}^{1/2}(\mathbf{x}^{1/2})\right) + \left(\hat{\mathbf{u}}^{1/2}(\mathbf{x}^{1/2}) - \hat{\mathbf{v}}^{1/2}(\mathbf{x}^{1/2})\right). && (22)
\end{aligned}$$

The first term is the error due to semi-Lagrangian advection,

$$\hat{\mathbf{u}}^{1/2}(\tilde{\mathbf{x}}^{1/2}) - \hat{\mathbf{u}}^{1/2}(\mathbf{x}^{1/2}) = -\Delta t^2 \underbrace{(\tfrac{1}{8}\nabla\hat{\mathbf{u}}^{1/2} \cdot \dot{\mathbf{u}})}_{\mathbf{e}^{1/2}} + O(\Delta t^3), \tag{23}$$

while the second is the error due to the discrepancy between $\hat{\mathbf{u}}^{1/2}$ and $\hat{\mathbf{v}}^{1/2}$,

$$\hat{\mathbf{u}}^{1/2}(\mathbf{x}^{1/2}) - \hat{\mathbf{v}}^{1/2}(\mathbf{x}^{1/2}) = -\Delta t^2 R[\mathbf{e}^0](\mathbf{x}^{1/2}) + O(\Delta t^3). \tag{24}$$

Together, these add up to

$$\hat{\mathbf{u}}_a^{1/2}(\mathbf{x}^1) - \hat{\mathbf{v}}_a^{1/2}(\mathbf{x}^1) = -\Delta t^2 \mathbf{e}^{1/2}(\mathbf{x}^1) + \Delta t^2 R[\mathbf{e}^0](\mathbf{x}^{1/2}) + O(\Delta t^3). \tag{25}$$

Now $\mathbf{e}^{1/2}$ can be taken to be evaluated at $\mathbf{x}^1$ instead, since this change only contributes another $O(\Delta t^3)$ term to the right-hand side. The total error of the reflection method is therefore

$$\begin{aligned}
\mathbf{u}^1 - \mathbf{v}^1 &= P[\hat{\mathbf{u}}_a^{1/2} - \hat{\mathbf{v}}_a^{1/2}] \\
&= \Delta t^2 P[\mathbf{e}^0 - \mathbf{e}^{1/2}] + O(\Delta t^3). && (26)
\end{aligned}$$

Since $\mathbf{e}^0 = \frac{1}{8}\nabla\mathbf{u}^0 \cdot \dot{\mathbf{u}}$ and $\mathbf{e}^{1/2} = \frac{1}{8}\nabla\hat{\mathbf{u}}^{1/2} \cdot \dot{\mathbf{u}}$, the difference between the two is $O(\Delta t)$. This makes the total error $O(\Delta t^3)$, proving the second-order accuracy of the method.