# A Two-Continua Approach to Eulerian Simulation of Water Spray

Michael B. Nielsen* Ole Østerby†

Aarhus University

## Abstract

Physics based simulation of the dynamics of water spray - water droplets dispersed in air - is a means to increase the visual plausibility of computer graphics modeled phenomena such as waterfalls, water jets and stormy seas. Spray phenomena are frequently encountered by the visual effects industry and often challenge state of the art methods. Current spray simulation pipelines typically employ a combination of Lagrangian (particle) and Eulerian (volumetric) methods - the Eulerian methods being used for parts of the spray where individual droplets are not apparent. However, existing Eulerian methods in computer graphics are based on gas solvers that will for example exhibit hydrostatic equilibrium in certain scenarios where the air is expected to rise and the water droplets fall. To overcome this problem, we propose to simulate spray in the Eulerian domain as a two-way coupled two-continua of air and water phases co-existing at each point in space. The fundamental equations originate in applied physics and we present a number of contributions that make Eulerian two-continua spray simulation feasible for computer graphics applications. The contributions include a Poisson equation that fits into the operator splitting methodology as well as (semi-)implicit discretizations of droplet diffusion and the drag force with improved stability properties. As shown by several examples, our approach allows us to more faithfully capture the dynamics of spray than previous Eulerian methods.

**CR Categories:** I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Physically based modeling

**Keywords:** animation, physics based animation, fluid simulation, multiphase flow, water spray simulation

**Links:** DL PDF

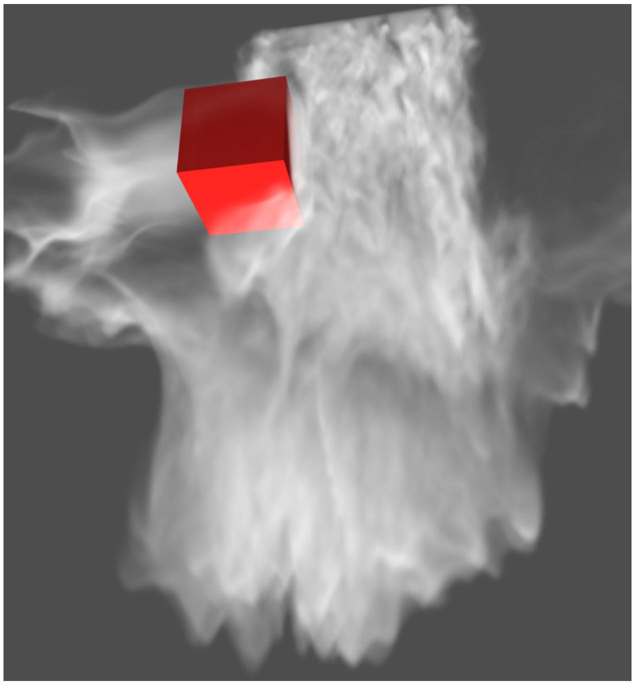

**Figure 1:** *Water droplets fall from a height of* $100m$ *and together with the air form a spray that interacts with a moving obstacle. Simulated using the Eulerian method proposed in this paper.*

## 1 Introduction

The problem of realistically and efficiently simulating water and air is of great importance to visual effects for feature films and video games. However, this problem continues to challenge researchers and practitioners in the computer graphics field. The sizable magnitude of the challenge is rooted partly in the wide range of scales at which water phenomena appear in nature - from deep water waves in the ocean to tiny droplets dispersed in the air. In computer graphics, specialized mathematical models and algorithms are applied to distinct phenomena to achieve the final fluid animation: from linear models of deep water waves synthesized in Fourier space to nonlinear models of turbulent topologically complex water, foam and spray simulated using a combination of Eulerian (volumetric) and Lagrangian (particle) methods.

Water spray - water droplets dispersed in air - occurs in nature as part of a wide range of phenomena: waterfalls, water jets, air interacting with waves and splashes to mention a few (Figure 2). The animation and visual effects industries are frequently faced with the challenge of modeling spray phenomena digitally and as such the problem of properly simulating spray is of great interest to the graphics community. However, simulating water spray turns out to be complex due to the large number of droplets, the variation in droplet size, and the fact that both the air and water phases must be accounted for to obtain the proper behavior. To accommodate this, several state of the art spray pipelines in the visual effects industry employ a combination of techniques [Geiger et al. 2006; Froemling et al. 2007]: Lagrangian particles are used for parts of the spray where individual droplets - or agglomerates thereof - are apparent, Eulerian grid methods are used for regions that do not exhibit a particulate look, and heuristic algorithms create plausible transitions between the two. Thus, often the qualitative (as opposed to quantitative) aspects will determine the choice between an Eulerian and Lagrangian method for modeling a particular spray phenomenon. Even though Eulerian spray simulation is an important part of industry water simulation pipelines, it typically requires a fair amount of tweaking to obtain visually plausible results in practice. In fact, Eulerian spray is currently simulated using gas solvers [Takahashi

*e-mail:nielsenmb@gmail.com

†e-mail:oleby@cs.au.dk

**ACM Reference Format**
Nielsen, M., Østerby, O. 2013. A Two-Continua Approach to Eulerian Simulation of Water Spray.
ACM Trans. Graph. 32, 4, Article 67 (July 2013), 10 pages. DOI = 10.1145/2461912.2461918
http://doi.acm.org/10.1145/2461912.2461918.

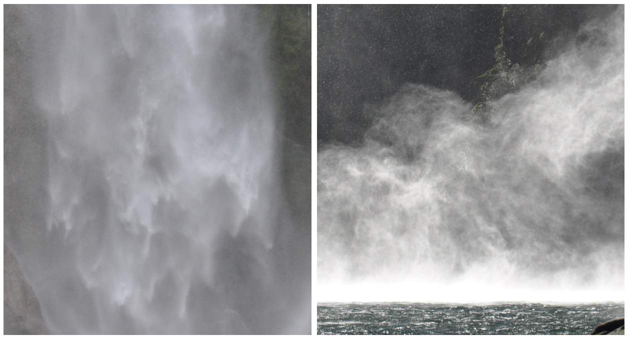

**Figure 2:** *Photographs of real spray phenomena. From left to right: (a) Waterfall (b) Spray creates an intricate pattern in the air.*

et al. 2003; Kim et al. 2006; Losasso et al. 2008] that fail to capture the correct dynamics in certain scenarios more faithfully captured by our method (Figure 3).

We propose a two-way coupled Eulerian spray simulation method based on a two-continua representation: water droplets and air are represented as two distinct continua that co-exist at each point in space. Each field - air and water - has its own volume fraction (the relative fraction of an infinitesimal volume it occupies) and velocity. Figure 2 illustrates real spray phenomena that we can to some extent capture qualitatively with our method: compare Figures 1 (lower part) and 2.a for the characteristic v-shaped patterns arising in waterfalls, as well as Figure 2.b with Figures 6 and 10 for the intricate patterns arising in evolving spray. The fundamental equations, theoretical justifications and assumptions behind the two-continua approach are not new - they have been developed and well studied in applied physics for more than a decade [Sirignano 1999; Brennen 2009]. We present a number of contributions that make numerical simulation of the two-continua approach feasible for computer graphics:

- A two-continua Poisson equation that fits into the operator splitting methodology
- A diffusion equation modeling droplet mass flux and an unconditionally stable discretization thereof
- A semi-implicit discretization of the drag force which exhibits improved stability over explicit discretization
- An Eulerian algorithm for simulating spray as a two-continua for computer graphics

We demonstrate and validate the properties of our model and algorithm with several examples. We focus solely on the dynamics of spray and refer to related work on how to handle transitions from water to spray, including how spray is generated from interfacial shear stress, jet breakup and other processes [Peachey 1986; Fournier and Reeves 1986; Sirignano 1999; Takahashi et al. 2003; Losasso et al. 2008; Mihalef et al. 2009; Brennen 2009].

## 2 Related Work

For an applied physics overview of the theory behind and fundamentals of spray computations, we refer the reader to the books by Sirignano [1999] and Brennen [2009]. Below we categorize related work in computer graphics according to how the droplets and the surrounding air is represented.

| symbol | description |
|---|---|
| $\alpha_\mathrm{a}$ | air volume fraction |
| $\alpha_\mathrm{w}$ | water droplet volume fraction |
| $\bar{\alpha}_\mathrm{w}$ | $\alpha_\mathrm{w}$ interpolated to voxel faces |
| $d$ | diffusion constant to control amount of diffusion |
| $\Delta t$ | time step of current iteration |
| $\Delta t_\mathrm{user}$ | time step specified by the user |
| $\Delta x$ | the side-length of a voxel |
| $C_D$ | drag coefficient |
| $\mathbf{f}_{a\leftarrow w}$ | interaction force density acting on air |
| $\mathbf{f}_{a\rightarrow w}$ | interaction force density acting on water droplets |
| $\mathbf{F}_{a\leftarrow w}$ | interaction force acting on air |
| $\mathbf{F}_{a\rightarrow w}$ | interaction force acting on water droplets |
| $\mathbf{g}$ | gravitational acceleration $\left((0,-9.81,0)\frac{m}{s^2}\right)$ |
| $k$ | time step number |
| $\nu_\mathrm{a}$ | kinematic viscosity of air $\left(1.51\cdot 10^{-5}\frac{m^2}{s}\right)$ |
| $p$ | pressure in spray |
| $r$ | droplet radius |
| $r_\mathrm{min}$ | minimum user-specified droplet radius |
| $r_\mathrm{max}$ | maximum user-specified droplet radius |
| $\mathrm{Re}$ | Reynolds number $(2r\left\|\mathbf{u}_\mathrm{rel}\right\|/\nu_\mathrm{a})$ |
| $\rho_\mathrm{a}$ | air density $\left(1.20\frac{kg}{m^3}\right)$ |
| $\rho_\mathrm{w}$ | water density $\left(997\frac{kg}{m^3}\right)$ |
| $\sigma_\mathrm{a}$ | CFL number for air advection step |
| $\sigma_\mathrm{w}$ | CFL number for water droplets |
| $\mathbf{u}_\mathrm{a}$ | air velocity $(\mathbf{u}_\mathrm{a}=(u_\mathrm{a},v_\mathrm{a},w_\mathrm{a}))$ |
| $\mathbf{u}_\mathrm{w}$ | water droplet velocity $(\mathbf{u}_\mathrm{w}=(u_\mathrm{w},v_\mathrm{w},w_\mathrm{w}))$ |
| $\mathbf{u}_\mathrm{rel}$ | relative velocity $(\mathbf{u}_\mathrm{a}-\mathbf{u}_\mathrm{w})$ |
| $\tilde{\mathbf{u}}_\mathrm{a}$ | air velocity after solving for advection and forces |
| $\tilde{\mathbf{u}}_\mathrm{w}$ | water velocity after solving for advection and forces |

**Table 1:** *Notation used throughout the paper.*

**Lagrangian Representation:** Peachey [1986] and Fournier and Reeves [1986] independently pioneered techniques for birthing spray particles from waves and for simulating the trajectories of the spray particles using Newtonian mechanics based on gravity. Fournier and Reeves included air drag as well. Several authors have proposed variations and combined with various methods for simulating the water [O'Brien and Hodgins 1995; Thürey et al. 2006; Chentanez and Müller 2010; Ihmsen et al. 2012]. As a general rule the spray particles do not interact, and the air flow - on which the drag force is based - is zero or procedurally generated. Thus these methods do not feature the two-way coupling with an air field included in our method. We stress however that our work remains orthogonal to and is intended to be used in combination with the Lagrangian spray methods.

**Eulerian Representation:** In this review we include previous work on simulating two-fluid mixtures as these methods can be used for spray simulation. Multiphase simulations - where air and water are separated spatially - enable the simulation of droplets dispersed in air [Losasso et al. 2006]. However, such an approach is typically not a feasible simulation method for water spray in computer graphics due to the large scale of scenes. To simulate cumuliform clouds (mixtures of air and water vapor), a single-phase fluid model is instead used by Miyazaki *et al.* [2002]. Mizuno *et al.* [2003] simulate two-fluid volcanic cloud mixtures using a similar single-phase gas solver augmented with a buoyancy force. Liu *et al.* [2008] recognize the need for a true two-fluid model for simulating tornados as a mixture of air and dust. However, separate mutually independent pressure solvers are applied to each of the phases which are coupled only through a drag force. A true variable density single-phase gas solver is used by both Kang *et al.* [2010] and Bao *et al.* [2010] to simulate miscible fluids under the assumption that the densities of the individual fluids do not change. All of these Eulerian methods are unable to model that spray is able to compress and

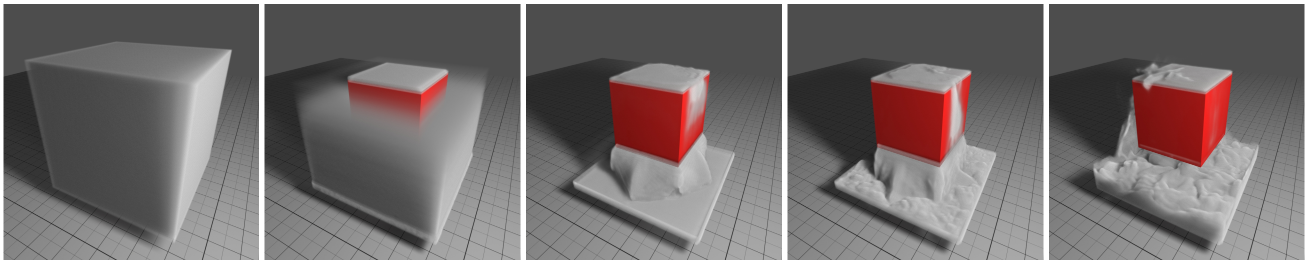

**Figure 3:** *Domain filled with a uniform water volume fraction of* 0.2*. Using our method, the air rises, and the droplets fall under gravity, compress at the bottom and are forced up under the internal obstacle. Existing Eulerian methods in computer graphics are all in hydrostatic equilibrium at the outset and retain the shape in the left-most image throughout the simulation (see the accompanying video).*

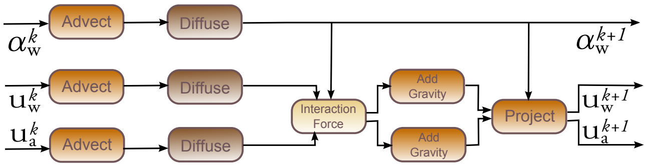
$\alpha_{\mathrm{w}}^k$
Advect
Diffuse
$\alpha_{\mathrm{w}}^{k+1}$
$\mathrm{u}_{\mathrm{w}}^k$
Advect
Diffuse
Interaction Force
Add Gravity
Project
$\mathrm{u}_{\mathrm{w}}^{k+1}$
$\mathrm{u}_{\mathrm{a}}^k$
Advect
Diffuse
Add Gravity
$\mathrm{u}_{\mathrm{a}}^{k+1}$

**Figure 4:** *Our solver uses an operator splitting approach.*

will exhibit hydrostatic equilibrium in certain cases where the air is supposed to rise and the droplets fall (Figure 3).

**Lagrangian Water Droplets and Eulerian Air/Mist Flow:** The first to present a combined Lagrangian-Eulerian method for simulating water spray in computer graphics are Takahashi *et al.* [2003]. Spray particles are emitted from high curvature water regions and affected by gravity and drag forces interpolated from the air flow of the underlying Eulerian air simulation. Mist is emitted from the spray particles and subsequently diffused and dissipated. Song *et al.* [2005] extend this work by proposing a method which features a two-way momentum transfer between the droplets and the air, and Kim *et al.* [2006] simulate spray as incompressible using droplets whose velocities are used as a force in a separate air simulation that passively advects mist emitted from the droplets. The method by Mihalef *et al.* [2009] also features two-way momentum transfer between water droplets and air, and in addition contributes with a more accurate drag force model. Losasso *et al.* [2008] argue that spray is compressible and propose a two-way coupled particle level set and diffuse SPH method that enables a smooth transition between ballistic droplet motion in the gravitational field and incompressible fluid behavior. A secondary one-way coupled air simulation is used to simulate the dynamics of mist sourced from the spray particles.
The Eulerian representations of air, spray and mist proposed by the methods in this category involve a gas solver. This implies that the Eulerian phase exhibits hydrostatic equilibrium for the example in Figure 3: the Eulerian mist will be in equilibrium, and while the Lagrangian droplets will fall, the air will also initially be in equilibrium - thus not giving rise to a physically plausible drag force.

# 3 Two-continua Method

Water droplets in a spray deform, collide and split up into smaller droplets through complex transformations [Villermaux and Boss 2009]. Here we adopt a simplified model in which droplets are modeled as spheres with varying radii. Instead of tracking individual droplets, we represent air and water volume fractions, $\{\alpha_{\mathrm{w}}, \alpha_{\mathrm{a}}\} \in [0;1]$, that denote the relative amount of water droplets and air in an infinitesimal volume (voxel in the discrete case) at each point. Each dependent variable in the dynamics equations is an instantaneous average over this infinitesimal volume that includes both water droplets and air. The basic assumption is that droplets are small compared to the voxel size, which will typically be true for computer graphics applications since we are simulating large scale scenes at relatively low resolution. The velocity of air $\mathbf{u}_{\mathrm{a}}$ and water droplets $\mathbf{u}_{\mathrm{w}}$ are typically different and hence stored separately at each point. We couple these velocities through an interaction force and a pressure projection. Our spray algorithm subjects the fundamental equations to an operator splitting approach similar to that introduced to graphics by Stam [1999]. Likewise our method involves advection, diffusion, force and pressure projection steps resulting from conservation of mass and momentum (Figure 4). All symbols used throughout the paper can be found in Table 1.

## 3.1 Conservation of Mass and Momentum

Ignoring thermodynamical effects, the dynamics of a two-continua mixture of water droplets and air is governed by the following equations that model the conservation of mass and momentum [Sirignano 1999; Brennen 2009]:

$$\alpha_{\mathrm{a}} + \alpha_{\mathrm{w}} = 1 \tag{1}$$

$$\frac{\partial}{\partial t}\left(\alpha_{\mathrm{a}}\rho_{\mathrm{a}}\right) + \nabla \cdot \left(\alpha_{\mathrm{a}}\rho_{\mathrm{a}}\mathbf{u}_{\mathrm{a}}\right) = 0 \tag{2}$$

$$\frac{\partial}{\partial t}\left(\alpha_{\mathrm{w}}\rho_{\mathrm{w}}\right) + \nabla \cdot \left(\alpha_{\mathrm{w}}\rho_{\mathrm{w}}\mathbf{u}_{\mathrm{w}}\right) = 0 \tag{3}$$

$$\alpha_{\mathrm{a}}\rho_{\mathrm{a}}\frac{D\mathbf{u}_{\mathrm{a}}}{Dt} = \alpha_{\mathrm{a}}\rho_{\mathrm{a}}\mathbf{g} + \mathbf{f}_{a\leftarrow w} - \alpha_{\mathrm{a}}\nabla p \tag{4}$$

$$\alpha_{\mathrm{w}}\rho_{\mathrm{w}}\frac{D\mathbf{u}_{\mathrm{w}}}{Dt} = \alpha_{\mathrm{w}}\rho_{\mathrm{w}}\mathbf{g} + \mathbf{f}_{a\rightarrow w} - \alpha_{\mathrm{w}}\nabla p \tag{5}$$

Eqs. (2) and (3) describe mass conservation of respectively the air and the water droplets, whereas Eqs. (4) and (5) describe conservation of momentum. We note that $\frac{D\mathbf{u}_{\mathrm{a}}}{Dt} \equiv \frac{\partial \mathbf{u}_{\mathrm{a}}}{\partial t} + (\mathbf{u}_{\mathrm{a}} \cdot \nabla)\mathbf{u}_{\mathrm{a}}$ and $\frac{D\mathbf{u}_{\mathrm{w}}}{Dt} \equiv \frac{\partial \mathbf{u}_{\mathrm{w}}}{\partial t} + (\mathbf{u}_{\mathrm{w}} \cdot \nabla)\mathbf{u}_{\mathrm{w}}$ are the material derivatives, and refer to Table 1 for the definition of the remaining symbols. Our contribution in this section is to re-formulate the mass conservation equations and derive a Poisson equation that fits into the operator splitting framework typically used in graphics. We first eliminate Eq. (1) and the unknown $\alpha_{\mathrm{a}}$ by defining $\alpha_{\mathrm{a}} \equiv 1 - \alpha_{\mathrm{w}}$. Next, we make the assumption that the densities of the two phases, $\rho_{\mathrm{a}}$ and $\rho_{\mathrm{w}}$, remain constant. This assumption precludes us from modeling velocities above the speed of sound and compression of air pockets entrapped in liquid, but allows us to divide the densities out of Eqs. (2) and (3). With these two simplifications, adding Eqs. (2) and (3) causes the time derivatives to cancel out and we obtain

$$\nabla \cdot \left(\alpha_{\mathrm{w}}\mathbf{u}_{\mathrm{w}} + (1 - \alpha_{\mathrm{w}})\mathbf{u}_{\mathrm{a}}\right) = 0 \tag{6}$$

This equation states that a weighted sum of the air and water velocity fields is divergence free, and allows us to model that air entering a voxel is able to push droplets out and vice versa. Note that when $\alpha_\mathrm{w}$ is zero or one, Eq. (6) reduces to the usual incompressibility constraint for air and water respectively. Assuming an operator splitting approach and using a forward Euler discretization of the time derivative, the pressure projection step becomes

$$\mathbf{u}_\mathrm{a} = \tilde{\mathbf{u}}_\mathrm{a} - \frac{\Delta t}{\rho_\mathrm{a}}\nabla p \tag{7}$$

$$\mathbf{u}_\mathrm{w} = \tilde{\mathbf{u}}_\mathrm{w} - \frac{\Delta t}{\rho_\mathrm{w}}\nabla p \tag{8}$$

Inserting Eqs. (7) and (8) into Eq. (6) and re-arranging terms, we obtain the variable-coefficient Poisson equation

$$\nabla\cdot(\alpha_\mathrm{w}\tilde{\mathbf{u}}_\mathrm{w} + (1-\alpha_\mathrm{w})\tilde{\mathbf{u}}_\mathrm{a}) = \frac{\Delta t}{\rho_\mathrm{a}\rho_\mathrm{w}}\nabla\cdot[(\rho_\mathrm{w} - \alpha_\mathrm{w}(\rho_\mathrm{w} - \rho_\mathrm{a}))\,\nabla p] \tag{9}$$

which we solve in combination with Eqs. (3), (4) and (5). These equations enforce $\alpha_\mathrm{w} \in [0;1]$ for all $t > 0$ provided it holds at $t = 0$ and $\alpha_\mathrm{w}$ varies smoothly in time: $\alpha_\mathrm{w} = 0 \Rightarrow \partial\alpha_\mathrm{w}/\partial t \geq 0$ by Eq. (3) and $\alpha_\mathrm{w} = 1 \Rightarrow \partial\alpha_\mathrm{w}/\partial t \leq 0$ by Eqs. (3) and (6). To implicitly define a ground level, we furthermore define $p = p' - \rho_\mathrm{a}|\mathbf{g}|y$, where $p'$ is the true pressure and $y$ is the height [Bridson 2008]. In the case of the two-continua equations this means that $\alpha_\mathrm{a}\rho_\mathrm{a}\mathbf{g}$ cancels out from Eq. (4) and $\alpha_\mathrm{w}\rho_\mathrm{w}\mathbf{g}$ is replaced by $\alpha_\mathrm{w}(\rho_\mathrm{w} - \rho_\mathrm{a})\mathbf{g}$ in Eq. (5).

### 3.2 Droplet Diffusion

When the relative velocity between air and water $\mathbf{u}_\mathrm{rel}$ is sufficiently large, the interfacial shear stresses produce waves on the droplets that transition into smaller droplets transported into the air by turbulent motions. Empirical studies of annular flows in vertical pipes have shown that at the interface between air and water, the mass flux of droplets into the air region correlates with the square root of the interfacial shear stress [Brennen 2009, p. 236]. Since the stress is proportional to the velocity gradient for a Newtonian fluid, we represent the magnitude of the mass flux by $d\sqrt{|\mathbf{u}_\mathrm{rel}|}$. We then model the mass flux via a diffusion process operating on the volume fraction $\alpha_\mathrm{w}$ and the velocity $\mathbf{u}_\mathrm{w}$. We exemplify here with diffusion of the volume fraction. Velocity diffusion is done similarly component wise. We make the assumption that the flux of $\alpha_\mathrm{w}$ at a certain point in space will be in the direction of and proportional to the negative gradient of $\alpha_\mathrm{w}$: droplets are then transported from dense droplet regions into regions dominated by air and the magnitude of the flux depends on the difference in water droplet volume fraction. The flux density thus becomes

$$\mathbf{\Phi} = -d\sqrt{|\mathbf{u}_\mathrm{rel}|}\nabla\alpha_\mathrm{w} \tag{10}$$

Inserting this flux density into the general formulation for a conservation law, $\frac{\partial\alpha_\mathrm{w}}{\partial t} + \nabla\cdot\mathbf{\Phi} = 0$, results in the diffusion equation

$$\frac{\partial\alpha_\mathrm{w}}{\partial t} - \nabla\cdot\left[d\sqrt{|\mathbf{u}_\mathrm{rel}|}\nabla\alpha_\mathrm{w}\right] = 0 \tag{11}$$

which must be solved at every grid point.

### 3.3 Air and Water Droplet Interaction

We employ a macroscopic model to describe the two-way interaction forces between water droplets and air - which for the results in this paper is restricted to a drag force. The drag force is parallel to and depends non-linearly on the relative velocity between water droplets and air. This makes an explicit discretization of the drag force infeasible for computer graphics as it is subject to a stricter time step restriction than that imposed by the CFL condition. To combat this problem, we propose a semi-implicit discretization which in practice does not impose further time step restrictions (Figure 5). This discretization is not restricted to Eulerian methods but can be applied to the particles in a Lagrangian framework as well.

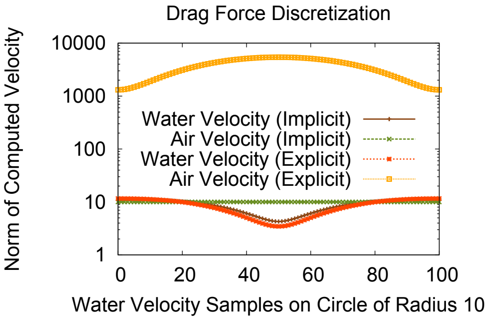

**Figure 5:** *The air- and water-velocity norms after application of drag forces (shown on a log-scale) for a constant initial air velocity of norm* $20m/s$*, initial water velocities of norm* $10m/s$ *distributed on a circle, a time step of* $0.005s$ (i.e. *an order of magnitude smaller than* $1/24s$) *and a droplet radius of* $0.1mm$*. The resulting velocity norms should be in the interval* $[0;20]$*. An explicit discretization of the drag force leads to over-shoots in the air velocity norm which remains bounded using our semi-implicit discretization.*

The drag force acting on a water droplet is [Mihalef et al. 2009]:

$$\mathbf{F}_{a\rightarrow w} = \frac{1}{2}C_D\rho_\mathrm{a}\pi r^2\,|\mathbf{u}_\mathrm{rel}|\,\mathbf{u}_\mathrm{rel} \tag{12}$$

where $C_D = 24/\mathrm{Re}^\gamma$. To represent water droplets with varying radii without introducing additional fields, we assume that the radius of a droplet correlates with $\alpha_\mathrm{w}$. This heuristic assumption is based on the intuition that - as the concentration of droplets increases - it is likely that droplets will collide and coalesce. In particular the radii of the droplets are computed at the voxel faces using linear interpolation based on the water volume fraction: $r = \bar{\alpha}_\mathrm{w}r_\mathrm{max} + (1-\bar{\alpha}_\mathrm{w})r_\mathrm{min}$. Based on experimental observations [Sirignano 1999], $\gamma = 1.0$ for $\mathrm{Re} \leq 1$, $\gamma = 0.72$ for $\mathrm{Re} > 30$ and linear interpolation can be used for Reynolds numbers in-between. To turn Eq. (12) into a force density required by the two-continua equations, we divide by the volume of a spherical particle $\frac{4}{3}\pi r^3$ and multiply by the volume fraction $\alpha_\mathrm{w}$ (to average over the total volume of the voxel as opposed to over the volume occupied by water droplets), thus obtaining:

$$\mathbf{f}_{a\rightarrow w} = \beta\,|\mathbf{u}_\mathrm{rel}|^{1-\gamma}\,\mathbf{u}_\mathrm{rel} \tag{13}$$

where $\beta = \frac{9}{2\gamma}\alpha_\mathrm{w}\rho_\mathrm{a}\nu_\mathrm{a}^\gamma r^{-(1+\gamma)}$. To conserve momentum, the force density $\mathbf{f}_{a\leftarrow w} = -\mathbf{f}_{a\rightarrow w}$ acts on the surrounding air, and when using operator splitting we need to solve the following two coupled ODEs at each grid point:

$$\frac{d\mathbf{u}_\mathrm{a}}{dt} = \mathbf{f}_{a\leftarrow w}/((1-\alpha_\mathrm{w})\rho_\mathrm{a}) = -\beta_\mathrm{a}\,|\mathbf{u}_\mathrm{rel}|^{1-\gamma}\,\mathbf{u}_\mathrm{rel} \tag{14}$$

$$\frac{d\mathbf{u}_\mathrm{w}}{dt} = \mathbf{f}_{a\rightarrow w}/(\alpha_\mathrm{w}\rho_\mathrm{w}) = \beta_\mathrm{w}\,|\mathbf{u}_\mathrm{rel}|^{1-\gamma}\,\mathbf{u}_\mathrm{rel} \tag{15}$$

where $\beta_\mathrm{a} = \beta/((1-\alpha_\mathrm{w})\rho_\mathrm{a})$ and $\beta_\mathrm{w} = \beta/(\alpha_\mathrm{w}\rho_\mathrm{w})$. Below we derive our semi-implicit discretization which decouples Eqs. (14) and (15) as well as the individual coordinate equations. If the reader is merely interested in the results of these derivations it is safe to skip the remainder of this section and consult Eqs. (16) and (17) for the $\gamma = 1$ case, in addition to appendix A for the $\gamma \neq 1$ case.

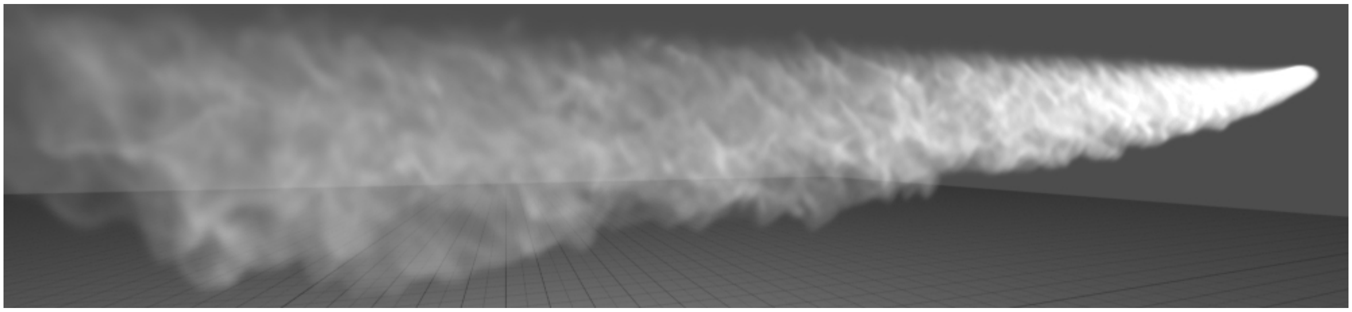

**Figure 6:** *A source moves from left to right over a distance of* $200m$ *and creates an intricate pattern of water droplets in its trail.*

The derivations proceed as follows: considering first the case $\gamma = 1$, we decouple Eqs. (14) and (15) by inserting the known values to the current time step $\mathbf{u}_{\mathrm{w}}^k$ and $\mathbf{u}_{\mathrm{a}}^k$ into Eqs. (14) and (15) respectively:

$$\begin{aligned}\frac{d\mathbf{u}_{\mathrm{a}}}{dt} &= -\beta_{\mathrm{a}}\left(\mathbf{u}_{\mathrm{a}} - \mathbf{u}_{\mathrm{w}}^k\right)\\ \frac{d\mathbf{u}_{\mathrm{w}}}{dt} &= \beta_{\mathrm{w}}\left(\mathbf{u}_{\mathrm{a}}^k - \mathbf{u}_{\mathrm{w}}\right)\end{aligned}$$

The analytical solutions at time step $k+1$ to these decoupled linear ODEs are given by:

$$\mathbf{u}_{\mathrm{a}}^{k+1} = \mathbf{u}_{\mathrm{w}}^k + e^{-\beta_{\mathrm{a}}\Delta t}(\mathbf{u}_{\mathrm{a}}^k - \mathbf{u}_{\mathrm{w}}^k) \quad (16)$$

$$\mathbf{u}_{\mathrm{w}}^{k+1} = \mathbf{u}_{\mathrm{a}}^k - e^{-\beta_{\mathrm{w}}\Delta t}(\mathbf{u}_{\mathrm{a}}^k - \mathbf{u}_{\mathrm{w}}^k) \quad (17)$$

Considering next the case $\gamma \neq 1$ we have a non-linear dependence on the relative velocity. To avoid notational clutter, we exemplify the solution approach for the $u$ component of the air velocity in $2D$. The approach is similar for all components of both the air and water velocities in $3D$ and the resulting expressions can be found in Appendix A. To decouple Eqs. (14) and (15) as well as the equations for the individual coordinates, we insert known values at time step $k$ into Eq. (14) as follows:

$$\begin{aligned}\frac{du_{\mathrm{a}}}{dt} &= -\beta_{\mathrm{a}}\left((u_{\mathrm{a}} - u_{\mathrm{w}}^k)^2 + (v_{\mathrm{a}}^k - v_{\mathrm{w}}^k)^2\right)^{\frac{1}{2}(1-\gamma)}\left(u_{\mathrm{a}} - u_{\mathrm{w}}^k\right)\\ &= \beta_{\mathrm{a}} f(u_{\mathrm{a}}) \qquad (18)\end{aligned}$$

Using a separation of variables approach [Coddington 1989] we solve Eq. (18) analytically. Multiplying by $dt$, dividing by $f(u_{\mathrm{a}})$ and integrating from $t$ to $t + \Delta t$, we obtain

$$\int_{u_{\mathrm{a}}(t)}^{u_{\mathrm{a}}(t+\Delta t)} \frac{du_{\mathrm{a}}}{f(u_{\mathrm{a}})} = \int_t^{t+\Delta t} \beta_{\mathrm{a}} dt \quad (19)$$

To obtain the solution $u_{\mathrm{a}}^{k+1}$ we re-arrange Eq. (19) and solve the resulting equation

$$\mathcal{F}(u_{\mathrm{a}}^{k+1}) = \mathcal{F}(u_{\mathrm{a}}^k) + \beta_{\mathrm{a}}\Delta t \quad (20)$$

where the right hand side is known and $\mathcal{F}$ is the antiderivative $\mathcal{F}'(u_{\mathrm{a}}) = \frac{1}{f(u_{\mathrm{a}})}$. It is easy to verify using Mathematica [Integrator 2013] that the analytical solution to the indefinite integral $\mathcal{F}(u_{\mathrm{a}}) = \int \frac{du_{\mathrm{a}}}{f(u_{\mathrm{a}})}$ evaluates to

$$\frac{1}{1-\gamma}(u_{\mathrm{w}}^k - u_{\mathrm{a}})^{(\gamma-1)} \times$$
$${}_2\mathrm{F}_1(\frac{1}{2}(1-\gamma), \frac{1}{2}(1-\gamma); \frac{1}{2}(3-\gamma); -\frac{(v_{\mathrm{a}}^k - v_{\mathrm{w}}^k)^2}{(u_{\mathrm{w}}^k - u_{\mathrm{a}})^2}) \quad (21)$$

where ${}_2\mathrm{F}_1$ is a hypergeometric function. To avoid expensive numerical root finding, we approximate the hypergeometric function ${}_2\mathrm{F}_1$ by its first order accurate Taylor expansion around the origin (the constant $1$) on the left hand side of Eq. (20). This is a reasonable approximation since ${}_2\mathrm{F}_1 = 1$ at the origin and decreases monotonically and slowly towards zero as its independent variable (the fourth argument) goes towards $-\infty$. Using this approximation we can solve Eq. (20) analytically and obtain

$$u_{\mathrm{a}}^{k+1} = u_{\mathrm{w}}^k - ((1-\gamma)\eta)^{\frac{1}{\gamma-1}} \quad (22)$$

where $\eta$ is the right hand side of Eq. (20). It is straightforward and relatively fast to evaluate Eq. (22), and an identical procedure for the remaining velocity components results in similar expressions which can be found in appendix A. Note that the right hand side of Eq. (22) is comprised of $u_{\mathrm{w}}^k$ minus a correction term, and hence exhibits the same structure as Eq. (16).

### 3.4 Implementation

This section describes the implementation and discretization details of the mathematical models. We employ a staggered grid representation with velocities stored at the voxel faces, and pressure and volume fractions stored at voxel centers [Fedkiw et al. 2001]. Algorithm 1 provides an overview of the individual steps of our method.

**Time Step Restrictions:** All steps of our spray algorithm - except for the advection steps employing the WENO scheme to discretize the convective derivatives (see below) - are based on (semi-)implicit discretizations. These discretizations have proven stable for all magnitudes of time steps tested. Hence the time step is restricted solely by the CFL condition imposed by the WENO scheme: $\Delta t = \min(\Delta t_{\mathrm{user}}, \sigma_{\mathrm{w}}\Delta x/\max(|\mathbf{u}_{\mathrm{w}}|))$. As will be elaborated below, the air velocity magnitude does not impact the overall time step of our algorithm.

**Water Volume Fraction Advection (Discretization of the Mass Conservation Equation (Eq. (3))):** In order to enable water droplets to compress and accumulate in a voxel it is important to use a conservative scheme for advecting the water droplet volume fractions. We employ a fifth order accurate WENO finite volume scheme in space combined with a third order accurate Runge Kutta in time (WENO+RK) [Jiang and Peng 1999; Shu 2009]. This ensures that the mass conservation property carries over to the discrete domain.

**Diffusion Discretization (Eq. (11)):** To ensure an unconditionally stable diffusion step we discretize implicitly in time using a first order accurate backward finite difference. We use second order accurate central difference approximations to the gradient and divergence operator. To obtain a compact seven-point stencil, we discretize $\nabla\alpha_{\mathrm{w}}$ at voxel faces, half-way between voxel centers. In

**Algorithm 1** Spray algorithm. See Table 1 for a definition of symbols, Figure 4 for an overview and section 3.4 for details.

```
while $t < t_{\text{end}}$ do
    $\Delta t$ = computeDt($\mathbf{u}_{\text{w}}^k$, $\sigma_{\text{w}}$, $\Delta t_{\text{user}}$)

    {Operations on water droplet volume fraction ($\alpha_{\text{w}}$):}
    $\tilde{\alpha}_{\text{w}}^k$ = advect($\alpha_{\text{w}}^k$, $\Delta t$)
    $\alpha_{\text{w}}^{k+1}$ = diffuse($\tilde{\alpha}_{\text{w}}^k$, $\Delta t$)
    $\bar{\alpha}_{\text{w}}^{k+1}$ = interpolateToVoxelFaces($\alpha_{\text{w}}^{k+1}$)
    ($m_{\text{a}}$, $m_{\text{w}}$) = computeMasks($\bar{\alpha}_{\text{w}}^{k+1}$)

    {Operations on velocity fields ($\mathbf{u}_{\text{a}}$ and $\mathbf{u}_{\text{w}}$):}
    $\tilde{\mathbf{u}}_{\text{w}}^k$ = advect($\mathbf{u}_{\text{w}}^k$, $\Delta t$, $m_{\text{w}}$)
    $\tilde{\mathbf{u}}_{\text{a}}^k$ = advect($\mathbf{u}_{\text{a}}^k$, $\Delta t$, $m_{\text{a}}$, $\sigma_{\text{a}}$)
    ($\mathbf{f}_{a \leftarrow w}$, $\mathbf{f}_{a \rightarrow w}$) = interactionForceDensity($\tilde{\mathbf{u}}_{\text{a}}^k$, $\tilde{\mathbf{u}}_{\text{w}}^k$, $\Delta t$, $\bar{\alpha}_{\text{w}}^{k+1}$)
    $\tilde{\mathbf{u}}_{\text{w}}^k$ = addForceDensity($\tilde{\mathbf{u}}_{\text{w}}^k$, $\Delta t$, $m_{\text{w}}$, $\mathbf{f}_{a \rightarrow w}$, $\mathbf{g}$)
    $\tilde{\mathbf{u}}_{\text{a}}^k$ = addForceDensity($\tilde{\mathbf{u}}_{\text{a}}^k$, $\Delta t$, $m_{\text{a}}$, $\mathbf{f}_{a \leftarrow w}$, $\mathbf{g}$)
    $\tilde{\mathbf{u}}_{\text{w}}^k$ = diffuse($\tilde{\mathbf{u}}_{\text{w}}^k$, $\tilde{\alpha}_{\text{w}}^k$, $\Delta t$)
    ($\mathbf{u}_{\text{w}}^{k+1}$, $\tilde{\mathbf{u}}_{\text{a}}^k$) = project($\tilde{\mathbf{u}}_{\text{w}}^k$, $\tilde{\mathbf{u}}_{\text{a}}^k$, $\bar{\alpha}^{k+1}$, $m_{\text{a}}$, $m_{\text{w}}$)
    $\mathbf{u}_{\text{a}}^{k+1}$ = clampToMaxVelRatio($\tilde{\mathbf{u}}_{\text{a}}^k$, $\mathbf{u}_{\text{w}}^{k+1}$, $m_{\text{a}}$, $m_{\text{w}}$)
    extrapolateIntoAirRegion($\mathbf{u}_{\text{w}}^{k+1}$, $m_{\text{w}}$)
    extrapolateIntoWaterRegion($\mathbf{u}_{\text{a}}^{k+1}$, $m_{\text{a}}$)

    {Other operations:}
    advanceSourcesAndObstacles($\Delta t$)
    saveFields($\mathbf{u}_{\text{a}}^{k+1}$, $\mathbf{u}_{\text{w}}^{k+1}$, $\alpha_{\text{w}}^{k+1}$)
    $t = t + \Delta t$
end while
```

1D we have

$$\frac{\alpha_{\text{w}_i}^{k+1} - \alpha_{\text{w}_i}^{k}}{\Delta t} - \frac{b_{i+1/2} \nabla \alpha_{\text{w}_{i+1/2}}^{k+1} - b_{i-1/2} \nabla \alpha_{\text{w}_{i-1/2}}^{k+1}}{\Delta x} = 0 \quad (23)$$

where $\nabla \alpha_{\text{w}_{i+1/2}}^{k+1} = (\alpha_{\text{w}_{i+1}}^{k+1} - \alpha_{\text{w}_i}^{k+1})/\Delta x$ and $b = d\sqrt{|\mathbf{u}_{\text{rel}}|}$ is computed halfway between grid points using linear interpolation. The above procedure generalizes coordinate wise to higher dimensions. At the border of the domain and at the surface of internal obstacles we set the flux in Eq. (10) to zero by enforcing the Neumann boundary condition $\nabla \alpha_{\text{w}} = 0$. We solve this linear system using a multigrid solver [Briggs et al. 2000]. A similar strategy is applied to each coordinate of $\mathbf{u}_{\text{w}}$ in the velocity diffusion step.

**Interpolation to Voxel Faces:** The discretized pressure projection step (see below) requires values of the volume fractions defined at voxel faces and we compute these using a second order accurate linear interpolation.

**Mask Computation:** To facilitate faster velocity advection and force steps, these are performed only at faces where the respective air and water masks are non-zero. In particular, masks are computed to identify which voxel faces contain fractions of water droplets and air. The water droplet mask, $m_{\text{w}}$, is set to 1 if $\alpha_{\text{w}} > 0$, otherwise it is set to 0. The air mask, $m_{\text{a}}$ is set to 0 if $\alpha_{\text{w}} \geq 1$ and set to 1 otherwise (see below for a discussion of why $\alpha_{\text{w}} \notin [0; 1]$ can occur in the discrete case).

**Velocity Advection and Forces (Discretization of the Momentum Conservation Equations (Eqs. (4) and (5))):** We apply the WENO+RK numerical scheme to each component of the water droplet velocity field. Because the magnitude of the air velocities can be significantly larger than the magnitude of the water droplet velocities, we currently solve the advection of air velocities using BFECC [Dupont and Liu 2007] combined with semi-Lagrangian advection. We iterate the air velocity advection step using the CFL number $\sigma_{\text{a}}$ until the air velocity has advanced forward by $\Delta t$. Ideally WENO should be employed for the air velocities as well, but in this case we favored speed over accuracy. The gravity force step is discretized using a first order forward Euler method in time, and the interaction forces are handled as described in section 3.3.

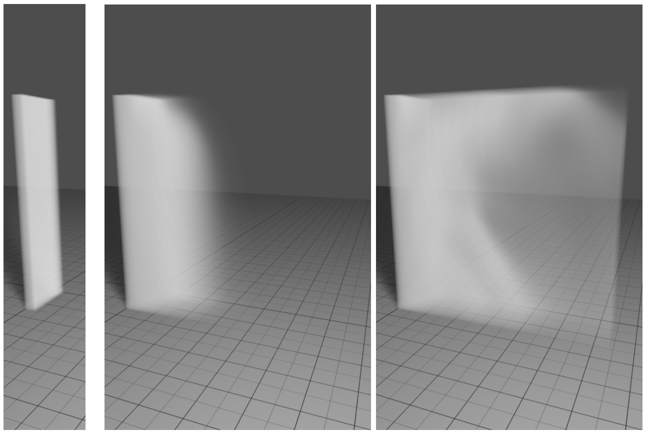

**Figure 7:** *Water droplets initially placed in a vertical bar and bordering still air (left) move downward with speeds $0.1m/s$ (middle) and $30m/s$ (right). The water droplets diffuse into the air region at a rate proportional to $\sqrt{|\mathbf{u}_{\text{rel}}|}$.*

**Pressure Projection (Discretization of the Poisson Equation (Eq. (9))):** The Poisson equation is discretized using second order accurate central finite differences of the gradient and divergence operators. The resulting linear system is solved using a multigrid solver [Briggs et al. 2000] combined with the penalization method for handling internal obstacles [Angot et al. 1999].

**Air Velocity Clamping:** Since the air density is orders of magnitude smaller than the water density, the coupled pressure solve can result in air velocities orders of magnitude higher than the velocity of water droplets, which is physically plausible. Typically this happens in areas under compression where $\alpha_{\text{a}} \ll \alpha_{\text{w}}$. In order to reduce the number of sub-frame iterations due to restrictive CFL conditions incurred by the air velocity, we clamp the air velocities to be within a user specified ratio relative to the maximum water droplet velocity. While this can introduce a small loss in volume (quantified in section 4), the visual deviation is qualitatively small (Figure 8) and thus feasible for computer graphics.

**Velocity Extrapolation:** To serve as boundary conditions for advection in the next iteration, velocity is extrapolated into regions where its corresponding phase is undefined. The extrapolation is performed using a fast sweeping method [Tsai et al. 2003] in the gradient direction of the distance from the air and water regions respectively.

**A Note on the Properties of the Discrete Water Droplet Volume Fractions:** The property $\alpha_{\text{w}} \in [0; 1]$ does not carry over to the discrete case: (1) the linear interpolation of $\alpha_{\text{w}}$ to voxel faces can cause a non-zero flux of water droplets into a voxel even though $\alpha_{\text{w}} = 1$; and (2) forward Euler time discretization as well as WENO is known to cause over- and under-shoots. Though remedies have been proposed in cases where the maximum principle is satisfied [Zhang and Shu 2011], these do not directly apply in the case of the two-continua equations. During simulation we allow $\alpha_{\text{w}} \notin [0; 1]$ as practice has shown that $\alpha_{\text{w}}$ remains bounded in the discrete case (section 4), and that clamping $\alpha_{\text{w}}$ to $[0; 1]$ results in mass conservation issues. We do however clamp its interpolation at voxel faces $\bar{\alpha}_{\text{w}}$ to $[0; 1]$ in order to ensure all steps of the algorithm

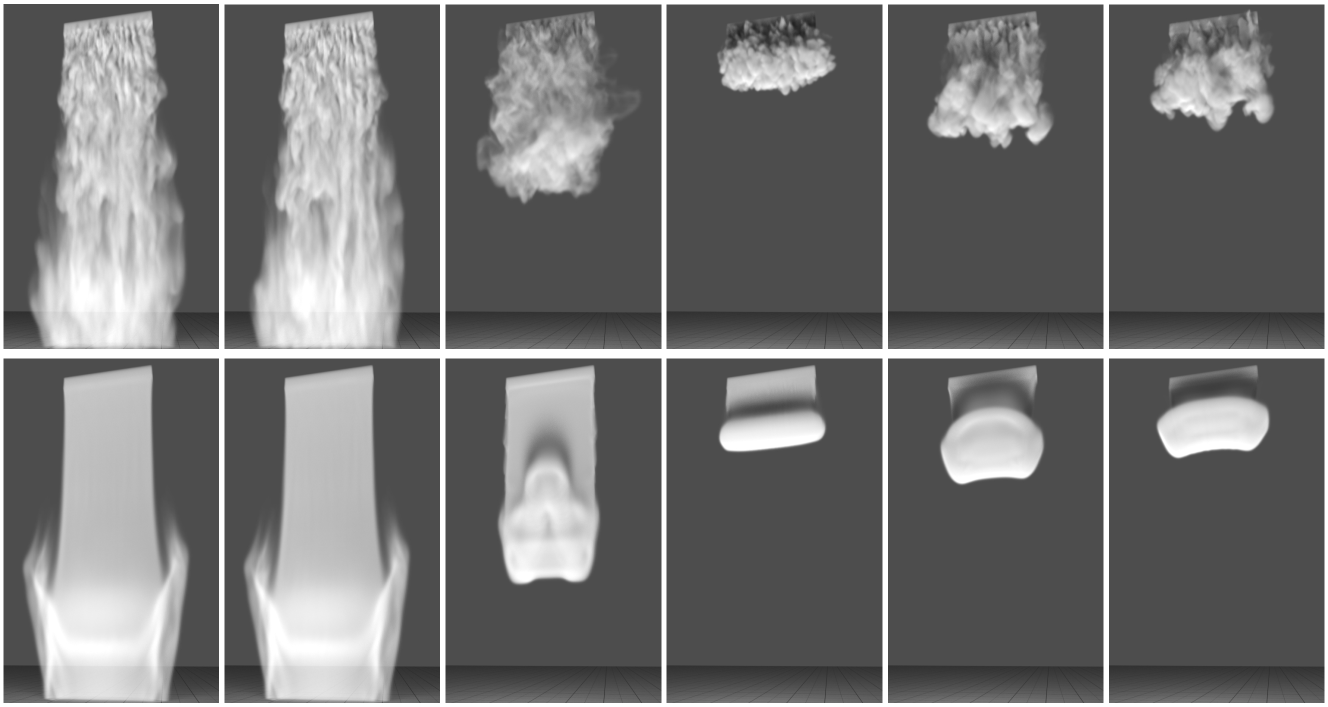

**Figure 8:** *Water droplets fall from a height of* $100m$. *Top row shows results obtained with vortex particles, and the bottom row without vortex particles (frame* 135 *shown in all cases). Identical input parameters (including vortex particle parameters) are used for all simulation methods (letters denote columns from left to right): (a) Our method without air velocity clamping (b) Our method with air velocity clamping (c) Variable density gas solver (d) Variable density gas solver with drag force (air velocity zero) (e) Constant density gas solver (f) Two-fluid model of Liu* et al.

are well behaved.

## 4 Results and Discussion

In this section we provide some background facilitating reproducibility of our examples as well as a discussion of our method and results. We refer the reader to the captions of Figures 1, 3, 6, 7, 8, 9, 10, and 11 for a description of the individual examples. Note that a few additional examples are available in the accompanying video. Table 2 lists properties of all simulations in this paper.

Our results are produced in the Windows operating system installed on a Mac Pro with 7GB of memory and two Intel Xeon quad-core 2.80GHz CPUs. We used the VS2005 compiler with full optimization enabled. The complete source code as well XML parameter files to reproduce all examples in this paper will be made available online [Nielsen 2013]. The images are produced by rendering $\alpha_{\mathrm{w}}$ with a ray marcher. For the examples in Figures 1, 6, 7, 8, 10, and 11 we use the open source vortex particle implementation of Pfaff *et al.* [2009] to emit vortex particles in the spray sources (between 100 and 400 particles each frame). For emission we follow the strategy of Selle *et al.* [2005] where vortex particles are seeded randomly with vorticity-direction orthogonal to the normal of the source and the main flow direction. Furthermore the vortex particles derive information from and act solely upon the water droplet velocity field. Note that vortex particles model bulk volume turbulence which does not accurately account for vorticity in mixtures of air and water droplets where the macroscopic behavior is influenced by the interface dynamics. Hence our use of vortex particles limits the realism achievable, but on the other hand presents a convenient way of breaking up the flow to force the underlying spray dynamics to come more into play. In general we recommend droplet radii ranging from $r_{\min} = 1\mu m$ to $r_{\max} = 1mm$ which is within the range of rain drop radii that have been observed at ground level [Villermaux and Boss 2009]. However, we use values of $r_{\max}$ up to $10cm$ in most examples (Figures 1, 3, 9, 10, 11) to exaggerate the difference in speed between small and large droplets. The assumption that a linear relationship exists between the water droplet volume fraction and the radii of water droplets in a voxel allows us to model varying droplet radii without introducing additional fields. However, it also prevents our method from accurately modeling scenarios where this assumption breaks down. It remains to be investigated for which visual phenomena this is the case.

With one exception we use a CFL number of $\sigma_{\mathrm{w}} = 0.9$ for all examples. The exception is Figure 9 where we conservatively chose a CFL number of 0.1 to obtain a more accurate and smooth result. For advection of air velocity we consistently use the CFL number $\sigma_a = 5$. A maximum air/water velocity norm ratio of 5 was chosen for all our examples, except Figure 8.a. In particular Figures 8.a and 8.b compare the results obtained using our method with and without air velocity clamping. The simulation time is roughly doubled without air velocity clamping, but qualitatively very little difference is observable. The remaining images in figure 8 illustrate the results obtained using existing methods for Eulerian spray simulation. For all methods a conservative WENO finite volume scheme is employed for the advection step. Furthermore identical input parameters are used except that for the previous methods, the gravity force suggested by Fedkiw *et al.* [2001] is used, and for Figure 8.d an air velocity of zero is assumed for drag force computation. Without vortex particles, the flow for all methods compared in figure 8 is predominantly laminar (bottom row), whereas the addition of vortex particles breaks up the flow and forces the underlying dynamics to come more into play (top row). We stress that identical input

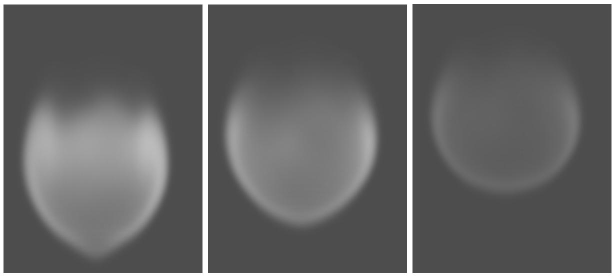

**Figure 9:** *A sphere of droplets with initial water volume fraction decreasing from left to right (*$1.0$*,* $0.6$ *and* $0.2$*) falls from a height of* $200m$*. The images shown are after 200 frames. As the droplet size and water volume fraction are correlated, this validates our ability to capture that larger droplets fall faster.*

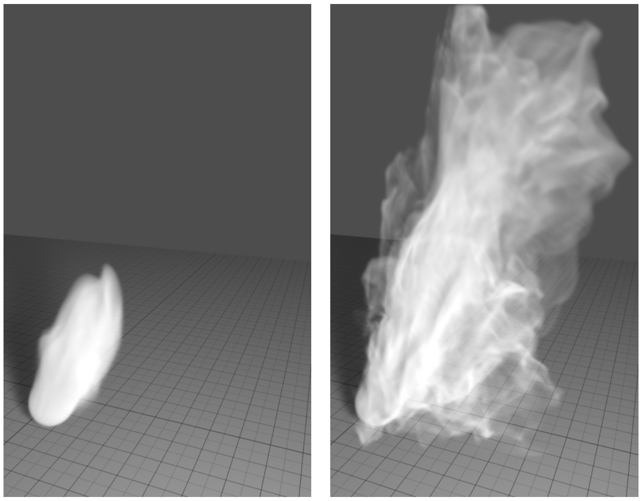

**Figure 10:** *A jet of droplets creates intricate patterns in the air.*

parameters are used for all competing Eulerian methods in figure 8 and that the difference apparent is solely due to how the various dynamical models respond differently to the same input parameters. The run-time of our spray solver is similar to that obtained using a variable-density single-phase gas solver (Figure 8.c). We note that the single-phase solvers and the two-fluid method of Liu *et al.* can be run at higher CFL numbers if resorting to unconditionally stable advection methods. The same may be true for our method, but this remains to be investigated. Since all parts of our solver - except for advection of $\alpha_\mathrm{w}$ and $\mathbf{u}_\mathrm{w}$ - are discretized (semi-)implicitly, the advection step is currently the most limiting in terms of time step restrictions. Recently a number of unconditionally stable schemes for conservative advection have been proposed [Lentine et al. 2011; Huang et al. 2012] and these hold promise to make our solver stable for CFL numbers above one.

Though the discrete version of our method does not limit $\alpha_\mathrm{w}$ to the interval $[0; 1]$, $\alpha_\mathrm{w}$ remains bounded throughout the simulation (Table 2). The highest compression ($\alpha_\mathrm{w} > 1$) is observed for the example in Figure 3. This is related to the relatively large deviation from zero in the divergence norm (cf. Table 2) of the combined velocity field after air velocity clamping and is due to the pressure developing at the bottom. In particular this results in high air velocities due to the density ratio between water and air. In the accompanying video a subtle jaggy motion can be observed for the same example in the falling spray as it thins out. We hypothesize this is due to the second order accurate linear interpolation of $\alpha_\mathrm{w}$ to the voxel faces, and alternatives should be investigated. Finally, artificial sticking to the internal obstacle is also apparent - a problem which could be alleviated by using the method of Batty *et al.* [2007].

| **fig** | **type** | **grid res** | $\alpha_\mathrm{w}$ | $\lvert\nabla\cdot\mathbf{u}\rvert_\infty$ | **avg time (frame/iter)** |
|---|---|---|---|---|---|
| 1 | s | $256\times256\times64$ | $[-0.3; 1.4]$ | $1\cdot10^{-6}$ | 1915 / 219 |
| 3 | s | $128\times128\times128$ | $[-0.7; 3.4]$ | $2\cdot10^{-1}$ | 505 / 115 |
| 6 | s | $512\times128\times64$ | $[0.0; 1.1]$ | $3\cdot10^{-7}$ | 274 / 154 |
| 7 | s | $128\times128\times32$ | $[0.0; 1.0]$ | $9\cdot10^{-7}$ | 262 / 90 |
| 8.a | s* | $256\times256\times64$ | $[-0.1; 1.9]$ | $3\cdot10^{-7}$ | 1200 / 232 |
| 8.b | s | $256\times256\times64$ | $[-0.1; 1.8]$ | $4\cdot10^{-7}$ | 660 / 127 |
| 8.c | v | $256\times256\times64$ | $[0.0; 1.0]$ | NA | 667 / 142 |
| 8.d | v | $256\times256\times64$ | $[0.0; 1.0]$ | NA | 490 / 131 |
| 8.e | c | $256\times256\times64$ | $[0.0; 1.0]$ | NA | 196 / 67 |
| 8.f | t | $256\times256\times64$ | $[0.0; 1.0]$ | NA | 531 / 185 |
| 9 | s | $64\times128\times64$ | $[-1.3; 2.6]$ | $1\cdot10^{-7}$ | 24 / 15 |
| 10 | s | $256\times256\times64$ | $[-0.4; 2.1]$ | $7\cdot10^{-5}$ | 1671 / 421 |
| 11 | s | $128\times256\times128$ | $[-0.2; 2.4]$ | $3\cdot10^{-4}$ | 1504 / 260 |

**Table 2:** *Properties of examples. The type denotes: (s) our spray algorithm, (s*) our spray algorithm without air velocity clamping, (v) variable density gas solver, (c) constant density gas solver, (t) the two-fluid model of Liu* et al. *The* $\alpha_\mathrm{w}$ *column gives the range of water volume fractions observed during the course of the simulation, and the* $\lvert\nabla\cdot\mathbf{u}\rvert_\infty$ *column gives the infinity norm of the average absolute divergence in a voxel of the combined velocity field* $(\alpha_\mathrm{w}\mathbf{u}_\mathrm{w} + (1-\alpha_\mathrm{w})\mathbf{u}_\mathrm{a})$ *after air velocity clamping. The last column lists the timings in seconds per frame and iteration.*

## 5 Conclusion and Future Work

The digital modeling of spray phenomena frequently challenges state of the art methods applied by the animation and visual effects industries. Taking outset in existing work in applied physics, we propose an Eulerian two-continua approach to spray simulation which we demonstrate is able to more faithfully capture the dynamics in certain scenarios where existing Eulerian methods fail. Our contributions include an algorithm for spray simulation which fits into the operator splitting framework popular in graphics as well as (semi-)implicit discretizations with increased stability behavior. We demonstrate the capabilities of our algorithm with several examples and are to some extent able to qualitatively capture characteristics of spray observed in photographs. We envision several exciting directions for future work that can leverage on the methods proposed in this paper: the development of turbulence models for spray, a closer integration of and coupling between Eulerian and Lagrangian spray simulation methods, the investigation of more accurate air-droplet interaction models, the application of our two-continua approach to other phenomena such as air-dust and bubble flows to mention a few. In addition, fluid control methods previously developed for gaseous simulations [Nielsen et al. 2009; Nielsen and Christensen 2010] should be investigated in the context of spray simulation. To facilitate future advancements in this area we make our source code publicly available for download.

## Acknowledgements

We thank Tim Ebling, Geoffrey Irving, Kevin Romond, Christoph Sprenger, Jonathan Swartz and Diego Trazzi for inspiring discussions, and are grateful to Peter Trier from the Alexandra Institute for providing a proprietary rendering framework and hardware. In addition we thank Louise Hellemose Rasmussen for the Milford Sound images (Figure 2), as well as Brian Bunch Christensen and Toshiya Hachisuka for commenting on drafts of the paper and providing help on the video. Finally we are grateful to the reviewers

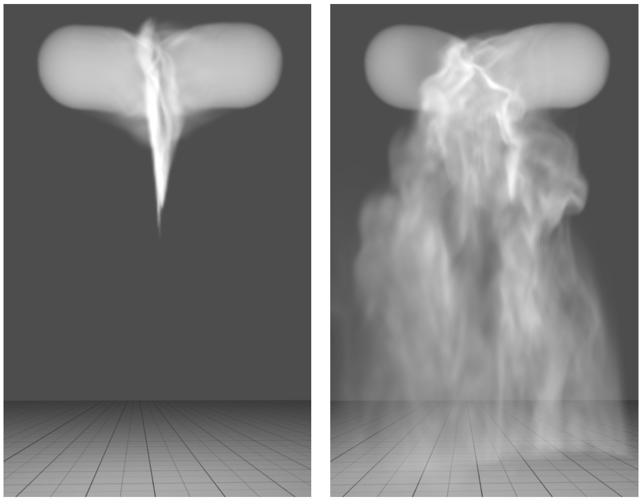

**Figure 11:** *Two spray sources with water volume fractions* 0.2 *collide which causes the spray to compress and fall faster.*

for valuable feedback and suggestions.

## References

ANGOT, P., BRUNEAU, C.-H., AND FABRIE, P. 1999. A penalization method to take into account obstacles in incompressible viscous flows. *Numerische Mathematik 81*, 4 (Feb.), 497–520.

BAO, K., WU, X., ZHANG, H., AND WU, E. 2010. Volume fraction based miscible and immiscible fluid animation. *Comput. Animat. Virtual Worlds 21*, 34 (May), 401–410.

BATTY, C., BERTAILS, F., AND BRIDSON, R. 2007. A fast variational framework for accurate solid-fluid coupling. In *ACM SIGGRAPH 2007 papers*, ACM, New York, SIGGRAPH '07.

BRENNEN, C. 2009. *Fundamentals of Multiphase Flow*. Cambridge University Press.

BRIDSON, R. 2008. *Fluid Simulation for Computer Graphics*. AK Peters.

BRIGGS, W. L., HENSON, V. E., AND MCCORMICK, S. F. 2000. *A multigrid tutorial (2nd ed.)*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA.

CHENTANEZ, N., AND MÜLLER, M. 2010. Real-time simulation of large bodies of water with small scale details. In *Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, Eurographics Association, Aire-la-Ville, Switzerland, SCA '10, 197–206.

CODDINGTON, E. A. 1989. *An Introduction to Ordinary Differential Equations*. Dover Books on Advanced Mathematics. Dover Publ.

DUPONT, T. F., AND LIU, Y. 2007. Back and forth error compensation and correction methods for semi-lagrangian schemes with application to level set interface computations. *Math. Comput. 76*, 258, 647–668.

FEDKIW, R., STAM, J., AND JENSEN, H. W. 2001. Visual simulation of smoke. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, ACM, New York, SIGGRAPH '01, 15–22.

FOURNIER, A., AND REEVES, W. T. 1986. A simple model of ocean waves. *SIGGRAPH Comput. Graph. 20*, 4 (Aug.), 75–84.

FROEMLING, E., GOKTEKIN, T., AND PEACHEY, D. 2007. Simulating whitewater rapids in ratatouille. In *ACM SIGGRAPH 2007 sketches*, ACM, New York, SIGGRAPH '07.

GEIGER, W., LEO, M., RASMUSSEN, N., LOSASSO, F., AND FEDKIW, R. 2006. So real it'll make you wet. In *ACM SIGGRAPH 2006 Sketches*, ACM, New York, SIGGRAPH '06.

HUANG, C.-S., ARBOGAST, T., AND QIU, J. 2012. An eulerian-lagrangian weno finite volume scheme for advection problems. *J. Comput. Phys. 231*, 11 (June), 4028–4052.

IHMSEN, M., AKINCI, N., AKINCI, G., AND TESCHNER, M. 2012. Unified spray, foam and air bubbles for particle-based fluids. *The Visual Computer 28* (Apr.), 669–677.

INTEGRATOR, 2013. http://integrals.wolfram.com.

JIANG, G.-S., AND PENG, D. 1999. Weighted eno schemes for hamilton–jacobi equations. *SIAM J. Sci. Comput. 21*, 6 (Dec.), 2126–2143.

KANG, N., PARK, J., YONG NOH, J., AND SHIN, S. Y. 2010. A hybrid approach to multiple fluid simulation using volume fractions. *Comput. Graph. Forum 29*, 2, 685–694.

KIM, J., CHA, D., CHANG, B., KOO, B., AND IHM, I. 2006. Practical animation of turbulent splashing water. In *Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation*, Eurographics Association, Aire-la-Ville, Switzerland, SCA '06, 335–344.

LENTINE, M., GRÉTARSSON, J. T., AND FEDKIW, R. 2011. An unconditionally stable fully conservative semi-lagrangian method. *J. Comput. Phys. 230*, 8 (Apr.), 2857–2879.

LIU, S., WANG, Z., GONG, Z., AND PENG, Q. 2008. Simulation of atmospheric binary mixtures based on two-fluid model. *Graph. Models 70*, 6 (Nov.), 117–124.

LOSASSO, F., SHINAR, T., SELLE, A., AND FEDKIW, R. 2006. Multiple interacting liquids. In *ACM SIGGRAPH 2006 Papers*, ACM, New York, SIGGRAPH '06, 812–819.

LOSASSO, F., TALTON, J., KWATRA, N., AND FEDKIW, R. 2008. Two-way coupled sph and particle level set fluid simulation. *IEEE Transactions on Visualization and Computer Graphics 14*, 4 (July), 797–804.

MIHALEF, V., METAXAS, D. N., AND SUSSMAN, M. 2009. Simulation of two-phase flow with sub-scale droplet and bubble effects. *Comput. Graph. Forum 28*, 2, 229–238.

MIYAZAKI, R., DOBASHI, Y., AND NISHITA, T. 2002. Simulation of cumuliform clouds based on computational fluid dynamics. In *EUROGRAPHICS 2002 Short Presentations*, 405–410.

MIZUNO, R., DOBASHI, Y., CHEN, B.-Y., AND NISHITA, T. 2003. Physics motivated modeling of volcanic clouds as a two fluids model. In *Proceedings of the 11th Pacific Conference on Computer Graphics and Applications*, IEEE Computer Society, Washington, DC, USA, PG '03, 440–444.

NIELSEN, M. B., AND CHRISTENSEN, B. B. 2010. Improved variational guiding of smoke animations. *Comput. Graph. Forum 29*, 2, 705–712.

NIELSEN, M. B., CHRISTENSEN, B. B., ZAFAR, N. B., ROBLE, D., AND MUSETH, K. 2009. Guiding of smoke animations through variational coupling of simulations at different resolutions. In *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, ACM, New York, NY, USA, SCA '09, 217–226.

NIELSEN, M. B., 2013. https://code.google.com/p/two-continua-eulerian-spray-simulation/.

O'BRIEN, J. F., AND HODGINS, J. K. 1995. Dynamic simulation of splashing fluids. In *Proceedings of Computer Animation*, IEEE Computer Society, Washington, DC, USA, CA '95, 198–205.

PEACHEY, D. R. 1986. Modeling waves and surf. In *Proceedings of the 13th annual conference on Computer graphics and interactive techniques*, ACM, New York, SIGGRAPH '86, 65–74.

PFAFF, T., THUEREY, N., SELLE, A., AND GROSS, M. 2009. Synthetic turbulence using artificial boundary layers. In *ACM SIGGRAPH Asia 2009 papers*, ACM, New York, SIGGRAPH Asia '09, 121:1–121:10.

SELLE, A., RASMUSSEN, N., AND FEDKIW, R. 2005. A vortex particle method for smoke, water and explosions. In *ACM SIGGRAPH 2005 Papers*, ACM, New York, NY, USA, SIGGRAPH '05, 910–914.

SHU, C.-W. 2009. High order weighted essentially non-oscillatory schemes for convection dominated problems. *SIAM Review 51*, 1, 82–126.

SIRIGNANO, W. 1999. *Fluid Dynamics and Transport of Droplets and Sprays*. Cambridge University Press.

SONG, O.-Y., SHIN, H., AND KO, H.-S. 2005. Stable but nondissipative water. *ACM Trans. Graph. 24*, 1 (Jan.), 81–97.

STAM, J. 1999. Stable fluids. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, ACM Press/Addison-Wesley Publishing Co., New York, SIGGRAPH '99, 121–128.

TAKAHASHI, T., FUJII, H., KUNIMATSU, A., HIWADA, K., SAITO, T., TANAKA, K., AND UEKI, H. 2003. Realistic animation of fluid with splash and foam. *Comput. Graph. Forum 22*, 3, 391–400.

THÜREY, N., RÜDE, U., AND STAMMINGER, M. 2006. Animation of open water phenomena with coupled shallow water and free surface simulations. In *Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation*, Eurographics Association, Aire-la-Ville, Switzerland, SCA '06, 157–164.

TSAI, R. Y.-H., CHENG, L.-T., OSHER, S., AND ZHAO, H.-K. 2003. Fast sweeping algorithms for a class of hamilton-jacobi equations. *SIAM Journal on Numerical Analysis 41*, 673–694.

VILLERMAUX, E., AND BOSS, B. 2009. Single-drop fragmentation determines size distribution of raindrops. *Nature Physics*.

ZHANG, X., AND SHU, C.-W. 2011. Maximum-principle-satisfying and positivity-preserving high order schemes for conservation laws: Survey and new developments. *Proceedings of the Royal Society A 467*, 2752–2776.

## A General Formulation of Drag Force

Referring to Table 1 and section 3.3 for a definition of the symbols not defined here, a general formulation of our drag force from section 3.3 in the case $\gamma \neq 1$ is as follows. Let the indices $p \in \{\mathrm{a}, \mathrm{w}\}$, $q = \{\mathrm{a}, \mathrm{w}\}\backslash p$ and $i \in \{x, y, z\}$. Furthermore let $\mathbf{u}_{p,i}$ denote the $i$'th component of velocity $\mathbf{u}_p$. Then the drag force is

$$\mathbf{u}_{p,i}^{k+1} = \mathbf{u}_{q,i}^{k} - (\tau \eta_{p,i})^{-\frac{1}{\tau}}$$

where

$$\begin{aligned}
\tau &= 1 - \gamma \\
\eta_{p,i} &= \frac{1}{\tau} \psi_i{}^{-\frac{1}{2}\tau} \left(\mathrm{F} + \beta_p \Delta t\right) \\
\mathrm{F} &= {}_2\mathrm{F}_1\left(\frac{1}{2}\tau, \frac{1}{2}\tau; \frac{1}{2}\tau + 1; -\frac{\phi_i}{\psi_i}\right) \\
\phi_i &= \sum_{j \in \{x,y,z\} \backslash i} \left(\mathbf{u}_{p,j}^{k} - \mathbf{u}_{q,j}^{k}\right)^2 \\
\psi_i &= \left(\mathbf{u}_{p,i}^{k} - \mathbf{u}_{q,i}^{k}\right)^2
\end{aligned}$$