Vortical Inviscid Flows with Two-Way Solid-Fluid Coupling

Mauricio Vines, Ben Houston, Jochen Lang, *Member*, *IEEE*, and Won-Sook Lee

Abstract—Vortex methods increasingly receive attention from the computer graphics community for simple and direct modeling of complex flow phenomena such as turbulence. The coupling between free-form solids, represented by arbitrary surface meshes, and fluids simulated with vortex methods, leads to visually rich simulations. In this paper, we introduce a novel approach for simulating the interaction between solids and inviscid fluids for high-quality simulations using Lagrangian vortex particles. The key aspect of our method is simulating the creation of vorticity at a solid's surface. While previous vortex simulators only focus on modeling the solid as a boundary for the fluid, our approach allows the accurate simulation of two processes of visual interest. The first is the introduction of surface vorticity in the main flow as turbulence (*vortex shedding*). The second is the motion of the solid induced by fluid forces. We also introduce to computer graphics the concept of source panels to model nonturbulent flow around objects. To the best of our knowledge, this is the first work on two-way coupling of 3D solids and fluids using Lagrangian vortex methods in computer graphics.

Index Terms—Fluid simulation, vortex methods, solid fluid coupling

1 Introduction

Turbulence gives flow phenomena its visual interest but simulating complex turbulent flows is one of the greatest challenges in fluid simulation. Turbulence is common and it can be observed in rising smoke and in water flowing past a rock. Turbulent flow is characterized by its randomness and its rotational nature. A natural way to model such flows is to explicitly represent and simulate their vorticity, or tendency to spin. Lagrangian vortex methods have three key advantages over traditional grid methods: they enable the representation of unbounded flows at arbitrary scales, and with no numerical dissipation. Simulating this kind of flow in traditional grid-based solvers is extremely challenging due to the limitations imposed by grid size and resolution.

Vortex methods in computer graphics have been traditionally applied to increase the detail in Eulerian grid simulations [1], [2]. More recently, Lagrangian vortex methods have also been employed for high-quality smoke simulations [3], [4], [5]. We follow this development direction due to the advantages of Lagrangian vortex methods over grid methods to represent unbounded, highly detailed flows.

Current Lagrangian vortex simulators in computer graphics focus on flow evolution and treat solid objects only as boundaries for the flow. However, vortex methods are used to accurately calculate forces in mechanical engineering applications such as airfoils [6]. In engineering, both solid shape and flow are known in advance in the design process. In contrast, our aim is to compute forces on solid objects of arbitrary shape under arbitrary flows.

We propose a novel vortex method to simulate the full interplay of rigid solids and inviscid fluids with two-way coupling. Our method is motivated by *boundary layer* theory [7], which establishes that a thin layer of fluid adheres to a solid surface due to viscosity. Fluids with very low viscosity are modeled as inviscid everywhere, except at a solid boundary where the viscous effects cannot be ignored. This approach to represent an inviscid flow originates from the D'Alembert's paradox, which describes the fact that, contrary to physical observation, the net force acting on a solid immersed in an irrotational and inviscid flow is zero. For instance, a sphere in an inviscid constant uniform flow would experience zero drag which is clearly incorrect.

Our method is based on three main components. The first is an inviscid potential-flow simulation, which is described in Sections 3 and 4. Here, solid objects are modeled as obstacles for the flow. Vortex stretching, which is the process where flow vorticity changes due to flow evolution, is simulated using our novel stable method. The second component is a boundary layer model that is described in Section 5. The boundary layer is modeled as vorticity that is generated at the solid surface and injected into the main flow. The third and final component of our method is a fluid force model acting on solids described in Section 6. We compute fluid forces based on the pressure changes induced by the release of vorticity into the main flow from the solid surface [8].

Our novel contributions are the following:

- We develop a method for coupling solids and fluids using Lagrangian vortex methods based on the generation of vorticity at the solid surface.

- *M. Vines is with the University of Ottawa, Ottawa, ON K1N 6N5, and with Exocortex Technologies Inc., 100 Gloucester Street, Office 410, Ottawa, ON K2P 1T3, Canada. E-mail: mvine059@uottawa.ca.*
- *B. Houston is with Exocortex Technologies Inc., 100 Gloucester Street, Office 410, Ottawa, ON K2P 1T3, Canada.*
- *J. Lang and W.-S. Lee are with the School of Electrical Engineering and Computer Science, University of Ottawa, 800 King Edward Avenue, Ottawa, ON K1N 6N5, Canada.*

Manuscript received 20 July 2012; revised 30 Dec. 2012; accepted 5 June 2013; published online 21 June 2013.
Recommended for acceptance by J. Keyser.
For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org, and reference IEEECS Log Number TVCG-2012-07-0141.
Digital Object Identifier no. 10.1109/TVCG.2013.95.

- We introduce to computer graphics a novel inviscid flow model around arbitrary solid objects. This model is based on vortex particle methods, singular source distributions at solid surfaces and a physically based boundary layer model.
- We devise a novel stable method for solving vortex stretching in Lagrangian particle simulations.

We demonstrate the high quality of our results through simulation of commonly seen phenomena and comparison with state-of-the-art simulations from the literature.

2 Related Work

Fluid simulation in computer graphics has greatly matured since the late 1990s, with a strong focus on Eulerian grid-based methods [9], [10]. Lagrangian particle methods are introduced as an alternative to grid simulations, predominantly Smoothed Particle Hydrodynamics (SPH) [11], [12]. Vortex methods, although introduced in computer graphics early on [13], have not been applied in high-quality 3D simulations until relatively recently [3], [4].

Vorticity has been applied in grid methods to recover and enhance details lost due to grid resolution. In the vorticity confinement technique [1], dissipated vorticity is reintroduced into the simulation grid as an additional force term for the Navier-Stokes equations. Hybrid vortex particle and grid simulations have been employed to recover subgrid flow details [14], and to generate high-resolution grids [15]. Vortex methods using precomputed boundary layer data have also been used to introduce turbulence into flow due to solid obstacles in a grid-based method [2]. Random vorticity is employed to generate turbulence in areas of high-turbulent kinetic energy in grid methods [16].

Smoke simulation using vortex particles considers solid objects only as obstacles so far. Solid boundaries can be modeled as a vortex sheet that are enforced by solving a boundary integral equation and simulating vortex shedding [3]. However, this technique, in general, is not suitable for objects of general shape [6]. We introduce to computer graphics a method based on sources, which is more robust in the presence of complex solid geometries. Vortex filaments, i.e., curves that concentrate vorticity, have also been employed in smoke simulations. Enforcing free-slip boundary conditions with vortex rings has been achieved by mirroring vortices inside the object with opposite spin [4], or by solving a boundary integral equation for zero velocity flux through the surface [5].

More recently, a hybrid method for representing vortical flows using domain decomposition methods has been proposed [17]. Grids are defined around rigid and deformable solid objects, and free surfaces to enforce boundary conditions by solving a Poisson equation on the voxelized solid boundaries. Grids are coupled with a Lagrangian vortex particle simulation that produces fine flow details. Particles' vorticity and grid velocities are coupled by solving constraints on grid boundaries. In contrast, we do not employ a grid to model fluid-solid interaction. We simulate solid-fluid interaction directly from vortex particles and the polygonal definition of a rigid solid object.

None of the above vortex methods solve two-way solid fluid coupling but treat the solid only as a boundary for the fluid motion. Lagrangian vortex methods have also been employed to model hot buoyant smoke by incorporating a baroclinic term in the flow equations [18], [19].

Besides obtaining visually realistic representations of flows, vortex methods have been addressed from the scientific visualization perspective. For example, vortical flow visualization methods based on sampling fluid properties along pathlines have been developed for engineering analysis [20].

Most work in two-way solid fluid coupling in computer graphics has been done on Eulerian grids, where solid objects are voxelized [9], thereby facilitating boundary condition enforcement and spatial derivative computations. More accurate coupling has been achieved by solving a coupled system for fluid and solid motion with a variational approach [21]. This is extended by explicit computation of momentum exchange between solids and fluids [22]. Solid fluid coupling in SPH has been achieved by creating virtual particles at solid objects surfaces, reducing solid-fluid interaction to particle-particle interaction [23]. Boundary particles have also been employed to compute friction and drag by calculating relative contributions accounting for irregular particle distributions [24].

Forces exerted by fluids on solid objects and solid-fluid coupling have been studied in mechanical engineering and aerodynamics. This can be done by calculating pressure changes due to the creation of vorticity at a solid surface [8]. This method has been applied to 2D simulations, both Lagrangian [25] and Eulerian [26]. This technique has also been formalized and extended to 3D fluid simulations [27], and utilized in coupling solids and fluids in the design of parachutes [28] and propellers [29]. An alternative for coupling between solids and fluids is to simulate both as a single continuum in a hybrid vortex-grid simulation [30]. Here, solid boundaries are defined by a level set, and rigid body motion is enforced by adding a penalty term to the flow equations in the space occupied by a solid object. This approach has several disadvantages compared to ours as it relies on an underlying grid, therefore limiting the simulation space. Also, solid motion is found through solving and advancing the boundary level set. This is not necessary in our method as we compute solid motion directly on the polygonal mesh.

Forces on solid objects can also be computed from the variation of the moments of vorticity in the fluid [31]. This method's properties have been studied for the case of several vortex rings interacting with a solid object [32]. The moments of vorticity are global flow quantities and do not provide information on the local influence on different solid objects. Such information can be obtained by defining a control volume around the solid and computing vorticity flux through the control surface [33]. Defining an adequate control volume for different interacting solids may not be a trivial task and we compute the vorticity flux at the solid surface because it characterizes the local force exerted by the fluid on each solid object.

3 Inviscid Potential-Flow Model

We focus on the simulation of an unbounded, incompressible, inviscid homogeneous flow interacting with several rigid solid objects in three dimensions.

3.1 Vortex Equations and Representation

Vortex methods model the flow through its vorticity ω, which represents the flow's tendency to spin, defined as the curl of the velocity field:

$$\omega = \nabla \times \mathbf{u}. \tag{1}$$

Vorticity evolution in our scenario is characterized by a vorticity transport equation that can be obtained from the Euler equations. Under irrotational external forces (such as gravity), the vorticity transport equation is formulated as follows:

$$\frac{\partial \omega}{\partial t} = -(\mathbf{u} \cdot \nabla)\omega + (\omega \cdot \nabla)\mathbf{u}. \tag{2}$$

Here, $\mathbf{u}$ is the fluid velocity and ∇ is the vector of spatial partial derivatives. The first term on the right-hand side of (2) is a vorticity advection term, which dictates that vorticity is transported with the flow. The second term is known as *vortex stretching* and it models the vorticity change in both spin direction and magnitude.

Fluid incompressibility is modeled by the continuity equation that characterizes mass conservation and is formulated as follows:

$$\nabla \cdot \mathbf{u} = 0. \tag{3}$$

We assume that vorticity is concentrated on discrete particles [3], or simply *vortices*. Each vortex is identified by its position $\mathbf{z}_i$, and its vorticity ω_i, which is a vector defining a spin direction and magnitude.

In the following section, we discuss how to calculate the flow velocity given its vorticity, which is required to solve (2) and advance the simulation.

3.2 Velocity Field Computation

3.2.1 Biot-Savart Formula

The velocity field induced by a vorticity field ω on a fluid region is defined by the Biot-Savart Law, which is a solution of a Poisson problem determined by (1) and (3). This is formulated as follows:

$$\mathbf{u}(\mathbf{x}) = \frac{1}{4\pi} \int_{Fluid} \omega(\mathbf{z}) \times \frac{\mathbf{x} - \mathbf{z}}{\|\mathbf{x} - \mathbf{z}\|^3} d\mathbf{z}.$$

Assuming the vorticity is concentrated only on discrete vortices, the above integral is equivalent to the sum of the contributions of each vortex j as follows:

$$\mathbf{u}(\mathbf{x}) = \frac{1}{4\pi} \sum_j \omega_j \times \frac{\mathbf{x} - \mathbf{z}_j}{\|\mathbf{x} - \mathbf{z}_j\|^3}. \tag{4}$$

Notice that the summation terms in this equation are singular. We employ a nonsingular velocity formulation that enables enhanced simulation control as described in the following section.

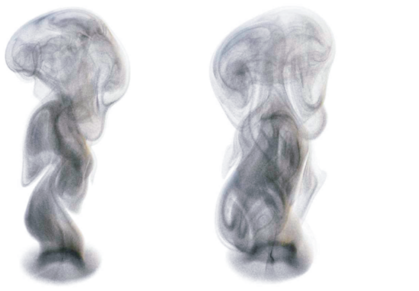

Fig. 1. Slow rising smoke with an emission radius of 1.0, simulated using the Rosenhead-Moore kernel (left), and our kernel with radius $\epsilon = 2.5$ (right). No significant visual quality degradation is observed.

3.2.2 Velocity Evaluation

We depart from the analytical model described above, and we employ a different approximation of the vorticity-induced velocity field. This approximation is not a solution of (1) and (3), however, as demonstrated in our examples, it leads to visually plausible flow simulations.

Instead of evaluating the Biot-Savart formula for calculating velocity, we employ a nonsingular function with finite influence radius, similarly to the approach in Angelidis et al. [4] for simulating vortex filaments. In their approach, such a kernel is used to simplify integration along 3D curves [34]. We calculate the velocity field through a formula of the form:

$$\mathbf{u}_j(\mathbf{x}) = (\omega_j \times (\mathbf{x} - \mathbf{z}_j)) w\big(\|\mathbf{x} - \mathbf{z}_j\|\big). \tag{5}$$

The velocity field must be divergence-free; otherwise, the condition of mass conservation does not hold. The following result characterizes functions that can be used as kernels that satisfy this property and we present its proof in Appendix A, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TVCG.2013.95.

Proposition 1. *For any continuous and differentiable radial basis function $w : \mathbb{R} \times \mathbb{R}$ and vectors ω, $\mathbf{x}$, and $\mathbf{z} \in \mathbb{R}^3$, the following holds: $\nabla \cdot [(\omega \times (\mathbf{x} - \mathbf{z})) w(\|\mathbf{x} - \mathbf{z}\|)] = 0$.*

We employ the following radial basis function w, with radius $\epsilon > 0$:

$$w(r) = \begin{cases} \left(1 - \frac{r^2}{\epsilon^2}\right)^3 & \text{if } |r| < \epsilon \\ 0 & \text{otherwise.} \end{cases} \tag{6}$$

Then, the velocity field in (5) is divergence-free and so is the sum of velocity induced by several vortices, i.e.,

$$\nabla \cdot \mathbf{u}(\mathbf{x}) = \nabla \cdot \sum_j (\omega_j \times (\mathbf{x} - \mathbf{z}_j)) w(\|\mathbf{x} - \mathbf{z}_j\|) = 0. \tag{7}$$

Using the kernel in (6) reduces the cost of computing velocity from $O(n^2)$ to $O(nk)$, where k is the amount of vortices within the radius ϵ of each particle. We evaluate (5) through a nearest neighbor search using a KD-tree structure.

We compare simulation results of a smoke column using two kernels in Fig. 1. We use our velocity definition and an approximation to the Biot-Savart formula known as the Rosenhead-Moore kernel [35], [18]. Since our kernel definition is different from that in the reference solution, our

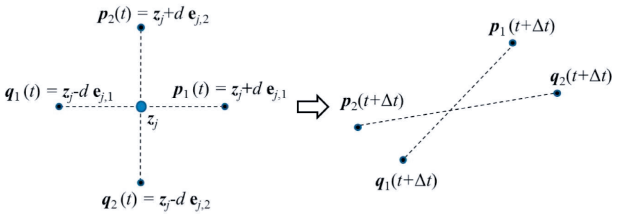
$p_2(t) = z_j + d\, \mathbf{e}_{j,2}$
$q_1(t) = z_j - d\, \mathbf{e}_{j,1}$
$p_1(t) = z_j + d\, \mathbf{e}_{j,1}$
z_j
$q_2(t) = z_j - d\, \mathbf{e}_{j,2}$
$p_1(t+\Delta t)$
$q_2(t+\Delta t)$
$p_2(t+\Delta t)$
$q_1(t+\Delta t)$

Fig. 2. Left: Sample markers and reference frame for vortex stretching. Right: Sample points after advection.

visual results are also different. Specific flow features can be reproduced by modifying the kernel radius. Evaluating the velocity field using finite kernels also enables further controlling the simulation results; a designer can limit the vortices' influence region on a complex scene.

With the velocity field we advance vortices in the flow, and compute vortex stretching, i.e., the change in vorticity spin and direction. In the following section, we present our approach to compute this term.

3.2.3 Vortex Stretching

To solve the vortex stretching term, a solution of the following equation has to be found:

$$\frac{\partial \omega}{\partial t} = (\omega \cdot \nabla)\mathbf{u} \equiv \nabla \mathbf{u} \cdot \omega.$$

Here, the gradient of velocity $\nabla \mathbf{u}$ is a tensor of rank two that is the transpose of the Jacobian matrix of the velocity $\mathbf{u}$. The vortex stretching term models the change of the vorticity vectors. Evaluation of this term may lead to instability due to an exponential increase in the vorticity magnitude [14].

We note that the gradient of velocity can be represented as follows:

$$\nabla \mathbf{u} = \frac{1}{2}\left(\nabla \mathbf{u} + (\nabla \mathbf{u})^T\right) + \frac{1}{2}\left(\nabla \mathbf{u} - (\nabla \mathbf{u})^T\right).$$

It can be shown that $\left(\nabla \mathbf{u} - (\nabla \mathbf{u})^T\right) \cdot \mathbf{b} = \omega \times \mathbf{b}$ for an arbitrary vector $\mathbf{b} \in \mathbb{R}^3$. Therefore, the vortex stretching term reduces to:

$$\nabla \mathbf{u} \cdot \omega = \frac{1}{2}\left(\nabla \mathbf{u} + (\nabla \mathbf{u})^T\right) \cdot \omega.$$

The matrix $\left(\nabla \mathbf{u} + (\nabla \mathbf{u})^T\right)$ corresponds to the strain tensor which is symmetric and it has the following eigendecomposition:

$$\left(\nabla \mathbf{u} + (\nabla \mathbf{u})^T\right) = \mathbf{Q}\mathbf{D}\mathbf{Q}^T,$$

where $\mathbf{Q}$ is an orthogonal matrix and $\mathbf{D}$ is a diagonal matrix. The strain tensor can be further expressed as $\mathbf{R}\mathbf{S}\mathbf{R}^T$, where $\mathbf{R}$ is a rotation matrix and $\mathbf{S}$ a scaling matrix. Then, solving vortex stretching reduces to computing a rotation and scaling for ω. Since vorticity magnitude may increase exponentially, we focus only on the rotational component of $\nabla \mathbf{u}$. The direct evaluation of $\nabla \mathbf{u}$ using singular particles may produce a nonsymmetric matrix, i.e., containing shearing as well. Instead, we employ a novel advection-driven method to approximate the rotation associated with the stretching term.

We approximate the rotation of the spin axis for each vortex by defining a local orthonormal frame of reference and calculating the rotation of this reference frame in a single, fictional advection step. Let $\mathbf{E}_j$ be the local orthonornal reference frame for vortex j. Then, we generate a set of marker particles located at a fixed distance d from the vortex position $\mathbf{z}_j$ along each axis in $\mathbf{E}_j$. We define $\mathbf{p}_{j,k} = \mathbf{z}_j + d\, \mathbf{e}_{j,k}$ and $\mathbf{q}_{j,k} = \mathbf{z}_j - d\, \mathbf{e}_{j,k}$ as shown in Fig. 2, where $\mathbf{e}_{j,k} \in \mathbf{E}_j$.

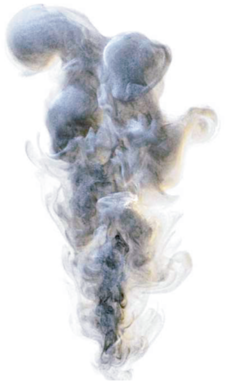

Fig. 3. Turbulent smoke column simulation with vortex stretching. Fast rotating smoke is obtained by generating vortices at different scales of strength and radius. Rotation is reduced by vorticity dissipation as the smoke rises, giving a more uniform appearance toward the top of the column.

Then, we advect marker particles at positions $\mathbf{p}$ and $\mathbf{q}$. From these advected positions, we obtain an advected reference frame given by the vectors:

$$\mathbf{b}_{j,k} = \mathbf{p}_{j,k}(t + \Delta t) - \mathbf{q}_{j,k}(t + \Delta t),$$

for $k \in \{1, 2, 3\}$. We assemble the advected axis $b_{j,k}$ in a new basis $\mathbf{B}_j$ from which we obtain a rotation matrix $\mathbf{R}_j$ using a polar decomposition. $\mathbf{R}_j$ is an orthogonalized basis for the vortex's advected frame of reference. Then, we update vorticity for each vortex particle j as follows:

$$\frac{\partial \omega_j}{\partial t} = \mathbf{R}_j \omega_j.$$

This is a stable vorticity update, and we show a turbulent smoke column example simulation using this method in in Fig. 3.

3.2.4 Vorticity Dissipation

Since the above vortex stretching method only introduces a change in the vorticity orientation and not in the magnitude, a turbulent energy cascade is not correctly represented. A fluid that preserves its energy at all scales would also be visually incorrect. We, therefore, employ a simple vorticity dissipation method that produces a visually realistic result.

We compute the *contained kinetic energy* [2] of a vortex i with influence radius ϵ as follows:

$$E_i = \int_{V_i} \frac{\rho}{2} \|\mathbf{u}_i(\mathbf{x})\|^2 dV_i,$$

where $\mathbf{u}_i$ and V_i are the vortex velocity and its region of influence, respectively. For our specific kernel, it can be shown that $E_i = k\|\omega\|^2 \epsilon^5$, where $k = \rho\pi \cdot 0.0107$.

Then, to preserve each vortex's energy, any change in the vorticity magnitude induces a change in the vorticity radius given by:

$$\epsilon_{new} = \epsilon_{old} \sqrt[5]{\frac{\|\boldsymbol{\omega}_{old}\|^2}{\|\boldsymbol{\omega}_{new}\|^2}}.$$

We apply this formula to recalculate the radius of each vortex as its vorticity magnitude changes.

3.3 Solution of the Vorticity Transport Equation

We evolve our simulation through time by solving (2) using the velocity defined in (7). We split (2) into each of its terms and solve them sequentially. We advect particles employing a second order Runge-Kutta time integration of velocity and we solve vortex stretching using our method in the previous section.

We can now summarize the advantages of vortex methods: First, they allow the simulation of fine details just by defining the vorticity of each particle according to the required detail scale. Second, only a sparse amount of data, in our case a set of particles, is required to produce such simulations. Lagrangian vortex simulations have no numerical dissipation, which on the other hand is a major issue in modeling inviscid flows using Eulerian grids. Finally, the pressure term disappears from the vorticity transport equation. This eliminates the need for computing a large matrix for pressure.

In the next sections, we detail the interaction between solids and fluids through boundary conditions.

4 Inviscid Solid Boundary Conditions

In our simulation, we assume a solid object is represented by a non-self-intersecting polygonal surface. Simulating inviscid flow around solid objects is achieved by enforcing a no-penetration boundary condition [3] at the solid surface which is formulated as $\mathbf{u}_{Solid} \cdot \hat{\mathbf{n}} = \mathbf{u}_{Fluid} \cdot \hat{\mathbf{n}}$.

In previous literature [3], a surface vorticity distribution tangent to the boundary is employed. The correct vorticity distribution to cancel flow through the surface is found by solving a system of $3N$ equations, where N is the number of surface elements. As opposed to their work, we introduce a novel method to computer graphics where a source distribution on the surface cancels the normal component of velocity. Only N equations need to be solved, with enhanced robustness in the presence of complex obstacles [6].

4.1 No-Penetration Condition

4.1.1 Source Sheets

We cancel the normal component of the flow at the surface by modeling the surface as a *source sheet* [6] and adding its induced velocity field to the flow. A source sheet can be seen as the limit of infinitely many sources (or sinks) distributed on a surface $\mathcal{S}$. The velocity induced by a source sheet, at a point in space $\mathbf{x}$, which does not lie on the sheet is defined as

$$\mathbf{u}(\mathbf{x}) = \int_{\mathcal{S}} \frac{\lambda(\mathbf{z})}{4\pi} \frac{\mathbf{x} - \mathbf{z}}{\|\mathbf{x} - \mathbf{z}\|^3} d\mathbf{z}. \tag{8}$$

Here, $\lambda(\mathbf{z})$ is the scalar source strength. The velocity at a point $\mathbf{x}_B$ that lies on the sheet is determined by a limiting process and it corresponds to $\lambda(\mathbf{x}_B)\hat{\mathbf{n}}/2$.

Below we show how to use source distributions to enforce no-penetration boundary conditions.

4.1.2 Boundary Integral Problem

Flow penetration through the surface occurs when $\Delta\mathbf{u} \cdot \hat{\mathbf{n}} = (\mathbf{u}_{Fluid} - \mathbf{u}_{Solid}) \cdot \hat{\mathbf{n}} \neq 0$. This quantity is canceled by the source sheet. The source sheet is determined by solving the unknown strength λ in the following boundary integral equation at surface locations $\mathbf{x}_B$:

$$\begin{aligned} -(\Delta\mathbf{u} \cdot \hat{\mathbf{n}})(\mathbf{x}_B) = {} & \frac{\lambda(\mathbf{x}_B)}{2} \\ & + \int_{\mathcal{S}} \frac{\lambda(\mathbf{z})}{4\pi} \frac{\mathbf{x}_B - \mathbf{z}}{\|\mathbf{x}_B - \mathbf{z}\|^3} d\mathbf{z} \cdot \hat{\mathbf{n}}(\mathbf{x}_B). \end{aligned} \tag{9}$$

The above equation corresponds to (8) evaluated at surface points $\mathbf{x}_B$. Equation (9) is discretized and solved as explained in the following section.

4.1.3 Discretization of Boundary Equations

We discretize (9) to solve the unknown scalar strength λ on the solid surface employing the *panel method* [36]. Each polygon or *panel* of the solid boundary corresponds to a constant strength source sheet that approximates the solution of (9).

At each panel i we define a control point $\mathbf{x}_i$ at the panel center. Then, we both sample the fluid velocity, and compute the influence of other panels at x_i.

We employ the finite kernel described in Section 3.2.2 to compute panel influence as follows:

$$\mathbf{u}_j^*(\mathbf{x}) = \int_j (\mathbf{x} - \mathbf{z}) w(\|\mathbf{x} - \mathbf{z}\|) d\mathbf{z}. \tag{10}$$

Since strength λ is constant for each panel j, the influence of panel j on the control point of panel i is simply $\lambda_j \mathbf{u}_j^*(\mathbf{x}_i)$. We rewrite (9) as follows:

$$-\Delta\mathbf{u}(\mathbf{x}_i) \cdot \hat{\mathbf{n}}_i = \frac{\lambda_i}{2} + \sum_{j=1}^{N_p} \lambda_j \mathbf{u}_j^*(\mathbf{x}_i) \cdot \hat{\mathbf{n}}_i. \tag{11}$$

Here, N_p is the number of panels. Equation (11) is a $N_p \times N_p$ linear system of the form $\mathbf{A}\boldsymbol{\lambda} = \mathbf{b}$. We evaluate the integrals for $\mathbf{u}_j^*$ using Gauss-Legendre quadratures. The matrix $\mathbf{A}$ is not necessarily symmetric, nor positive-definite and the kernel radii used to evaluate the matrix coefficients determine the sparsity of $\mathbf{A}$. For a rigid body, $\mathbf{A}$ is constant and we calculate it and invert it in a precomputation step.

Influence radii ϵ of each panel are determined from a *characteristic length* L_C of the solid object. In our simulations, the characteristic length employed corresponds to the radius of the smallest cylinder that contains the object and we use $0.5L_C \leq \epsilon \leq 1.5L_C$.

Once the source strength λ is determined for each panel, the solid object induced velocity is the sum of the velocities induced by each panel, i.e.,

$$\mathbf{u}_\lambda(\mathbf{x}) = \sum_{j=1}^{N_p} \lambda_j \mathbf{u}_j^*(\mathbf{x}). \tag{12}$$

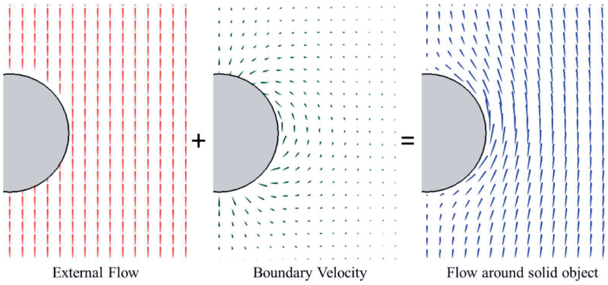

Fig. 4. Velocity fields for boundary conditions around a sphere. Left: External uniform flow. Center: Surface velocity field that cancels penetration. Right: Result of adding the external flow and the surface velocity satisfying the no-penetration boundary condition.

We add $\mathbf{u}_\lambda$ to the simulation, satisfying the no-penetration boundary condition, as shown in Fig. 4. By using our finite kernel, the effects of source panels are not evaluated in points that are too far from the solid object, where the induced velocity is negligible.

A comparison of no-penetration enforcement using our kernel and the Rosenhead-Moore kernel is shown in Fig. 5. We observe that both results are visually different, although both enforce no-penetration on the solid. The differences appear due to an overestimation of the velocity field induced by the employed kernel. By Proposition 1, different kernels can be applied to obtain a variety of visual results.

4.2 The Case of Finite Kernel Vortices

The source panel method presents additional challenges when using finite kernels for velocity. Consider a symmetric regular vortex jet toward a sphere as in Fig. 5. Small kernel radii introduce artifacts since: 1) Not all the solid object's panels are influenced by a small radius vortices, and 2) distances between vortices increase due to the insertion of the solid, reducing the mutual influence of particles. As vortices approach the solid object, general flow features are distorted by artificially dominant local flow structures. The exact error magnitude is difficult to estimate as panels' strengths are determined by a large linear system and sources and vortex radii may differ.

Enforcing the no-penetration boundary condition is equivalent to forcing the solid boundary to be a flow *streamline*, i.e., a curve that is tangent to the flow velocity. We achieve this by using *vortex images*: when a solid boundary intersects the influence radius of a vortex, a second vortex with same radius and opposite spin is generated at the mirror position across the solid surface. We add velocities of vortices and their mirrors to ensure the flow is tangential to the surface.

Source panels are still necessary to account for the flow's irrotational features (e.g., a user-defined background flow), whereas vortices and their mirrors model the rotational component of the flow. Then, we employ a combination of source panels and vortex mirroring in our examples.

5 Boundary Layer Model

The problem with the velocity field obtained using source panels is that the net force calculated on the solid object is zero. This can be resolved by calculating the surface vorticity around the object and then directly computing the pressure as it is commonly done in airfoil design. However, this type of solution strategy depends on the airfoil shape and specific flow conditions, which do not hold for the turbulent flows and general solid shapes in which we are interested.

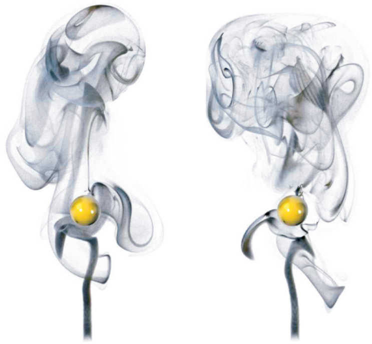

Fig. 5. Enforcing no-penetration boundary conditions on a unit radius sphere. Left: Source panels using the Rosenhead-Moore kernel. Right: Source panels using our kernel with radius $\epsilon = 1$.

Instead, we couple our potential-flow model with a boundary layer simulation. We impose an additional constraint on the tangential components of the flow velocity at the solid boundary. This allows simulating a variety of slip conditions where the no-slip boundary condition [37] is achieved by canceling the tangential component of flow velocity at the solid surface. This is in addition to the no-penetration boundary condition that constrains the normal component of the flow velocity at the solid surface. Fluid particles in the boundary layer are subject to a torque induced by the motion of other fluid particles farther from the solid surface. This torque translates into vorticity that is released into the flow as vortex shedding.

Then, we apply our boundary layer model in three steps: First, we determine the surface vorticity using a vortex sheet, whose induced velocity coincides with the slip velocity. Second, we enforce a slip boundary condition by emitting the surface vorticity into the main flow as new vortex particles. Third, we advect the newly created vortex particles to simulate boundary layer separation. We describe these steps in the following sections.

5.1 Surface Vorticity

Surface vorticity is found by modeling the solid boundary as a *vortex sheet*. This is the limit of infinitely many vortices lying on the surface. Surface vorticity γ is a vector quantity parallel to the solid surface [38]. This quantity is determined by a boundary integral equation whose structure is similar to that of source panels. The main difference is that γ has to be solved for both tangential components $\hat{\mathbf{s}}$ and $\hat{\mathbf{t}}$ of each panel's surface. We use the formulation by Park and Kim [3] and we refer the reader to their work for details. The boundary integral equation that solves unknown vorticity in the direction $\hat{\mathbf{s}}$ is the following:

$$\begin{aligned}(\Delta\mathbf{u}\cdot\hat{\mathbf{s}})(\mathbf{x}_B) = {} & \frac{(\gamma\times\hat{\mathbf{n}})(\mathbf{x}_B)}{2}\cdot\hat{\mathbf{s}}(\mathbf{x}_B) \\ & + \int_S \frac{\gamma(\mathbf{z})}{4\pi}\times\frac{\mathbf{x}_B-\mathbf{z}}{\|\mathbf{x}_B-\mathbf{z}\|^3}\,d\mathbf{z}\cdot\hat{\mathbf{s}}(\mathbf{x}_B).\end{aligned} \tag{13}$$

We discretize and solve (13) using our finite kernel as described in the next section.

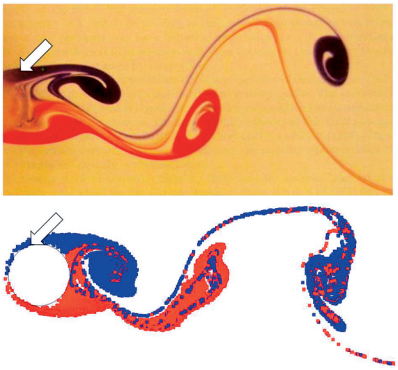

Fig. 6. Top: Kármán vortex street observed in flow past a cylinder (indicated with an arrow) from a real-life experiment in [39], A.E. Perry, M.S. Chong, T.T. Lim, "The vortex-shedding process behind two-dimensional bluff bodies," Journal of fluid Mechanics, Vol. 129, No. 2, pages 77-90, 1982, reproduced with permission. Bottom: Kármán vortex street from our 2D simulation on a unit radius sphere under a uniform flow, with $c_{Slip} = 1$ and kernel radius $\epsilon = 1.5$.

5.2 Discretization Using the Panel Method

We proceed with constant panels similar to Section 4.1.3 and define a control point $\mathbf{x}_i$ at the center of each panel i. Under these assumptions, we can rewrite (13) evaluated at the control points, using the velocity field $\mathbf{u}^*$ in (10) as follows:

$$\Delta\mathbf{u}(\mathbf{x}_i)\cdot\hat{\mathbf{s}}_i = \frac{\boldsymbol{\gamma}_i\times\hat{\mathbf{n}}_i}{2}\cdot\hat{\mathbf{s}}_i + \sum_j \boldsymbol{\gamma}_j\times\mathbf{u}^*_j(\mathbf{x}_i)\cdot\hat{\mathbf{s}}_i.$$

We represent γ in terms of the tangential components $\hat{\mathbf{s}}$ and $\hat{\mathbf{t}}$ of the surface, i.e., $\gamma = \gamma^s\hat{\mathbf{s}} + \gamma^t\hat{\mathbf{t}}$. Then, the above equation can be written as follows:

$$\begin{aligned}\Delta\mathbf{u}(\mathbf{x}_i)\cdot\hat{\mathbf{s}}_i = \frac{\gamma_i^t}{2} &+ \sum_j \gamma_j^s(\hat{\mathbf{s}}_i\times\hat{\mathbf{s}}_j)\cdot\mathbf{u}^*_j(\mathbf{x}_i)\\ &+ \sum_j \gamma_j^t(\hat{\mathbf{s}}_i\times\hat{\mathbf{t}}_j)\cdot\mathbf{u}^*_j(\mathbf{x}_i).\end{aligned} \tag{14}$$

The derivation of (14) is detailed in Appendix B, available in the online supplemental material. We set the panel influence radius to be the same used for source panels. Equation (14) is a linear system of equations of the form:

$$\mathbf{B}\begin{bmatrix}\gamma_i^s\\ \vdots\\ \gamma_i^t\\ \vdots\end{bmatrix} = \begin{bmatrix}\Delta\mathbf{u}(\mathbf{x}_i)\cdot\hat{\mathbf{s}}_i\\ \vdots\\ \Delta\mathbf{u}(\mathbf{x}_i)\cdot\hat{\mathbf{t}}_i\\ \vdots\end{bmatrix}. \tag{15}$$

The matrix $\mathbf{B}$ has dimensions $2N_p \times 2N_p$ as the influence of two components of vorticity per panel are considered. As in the case of the matrix for solving the no-penetration boundary conditions, in general the matrix $\mathbf{B}$ is not symmetric, nor positive definite but it will be sparse as long as the kernel radii in the velocity evaluation are small enough. We note that this matrix is again constant for a rigid solid object and we also calculate it and its inverse in a precomputation step.

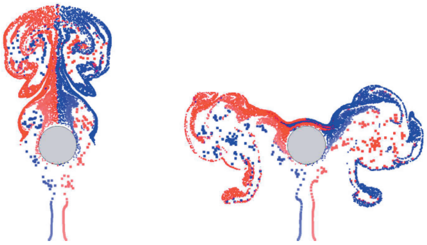

Fig. 7. Flow patterns of a vertical vortex jet around a static 2D unit radius sphere with kernel radius $\epsilon = 1$. Left: $c_{Slip} = 0$. Right: $c = 0.7$.

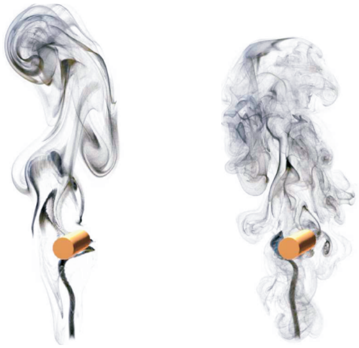

Fig. 8. Three-dimensional flow patterns of a vertical vortex jet around a static cylinder of unit radius. Kernel radius is $\epsilon = 1$ and $c_{Slip} = 0.0$ (left) and $c_{Slip} = 1.0$ (right).

Once the surface vorticity is determined for each panel, it is possible to model the development of the boundary layer which is discussed in the next section.

5.3 Vortex Shedding

We can enforce diverse slip boundary conditions through vortex shedding by emitting vortices in the flow. These can be used to cancel the slip velocity or only a part of it [3]. The velocity field induced by these vortices alters the velocity field around the object such that the boundary layer separates from the solid surface as shown in Fig. 6.

We apply a modified version of the method in [3]. We generate a new vortex at a close random distance d from each panel. The vorticity of each vortex is

$$\omega_i^{new} = c_{Slip}\gamma_i A_i \Delta t, \tag{16}$$

where γ_i and A_i correspond to panel i's vorticity and area, respectively. The slip coefficient c_{Slip} controls the friction between the solid and fluid. The free-slip boundary condition, where $c_{Slip} = 0$, occurs because the shed vortices have zero vorticity. Whereas for the no-slip boundary condition, where $c_{Slip} = 1$, the shed vortices have full strength. We set each vortex radius to the same radius used for panel calculations.

In previous literature [3], vortices are diffused into the main flow using a random walk. We do not use this strategy as in a nearly inviscid flow, viscosity is dominant only within the boundary layer. Outside this area, flow is dominated by advection, where flow separation from the solid boundary is the main phenomenon we focus on. We can reproduce natural boundary layer evolution using this very simple model as shown in Fig. 6, despite that our velocity computation formulation is different from that defined by the flow equations.

Comparison of different boundary layer separation patterns for different values of c_{Slip} are shown in Figs. 7 and 8. In addition to the physical representation of a flow past a solid object using vortex shedding, this feature adds great visual interest to the scenes as shown in Fig. 9.

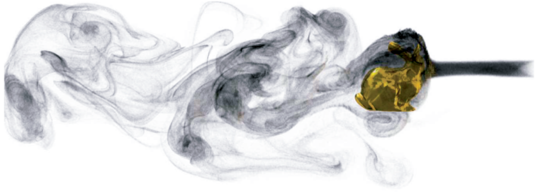

Fig. 9. Flow induced by a vortex jet past a static Stanford Bunny model with characteristic length $L_C = 0.75$ with $c_{Slip} = 1$ and kernel radius = 0.5.

6 Force Model

So far we have discussed the simulation of an inviscid flow around a solid object, to which we add the viscous effects of the boundary layer at the solid surface. This viscous boundary model allows us to simulate two-way solid fluid coupling because fluid forces acting on a solid are a consequence of the generation and evolution of the boundary layer [7]. More precisely, fluid forces acting on solids have been related to the vorticity flux from the solid surface into the main fluid [8], [27]. We base our approach on the work of Wu and Wu [27] as outlined below.

6.1 Surface Forces

The force exerted by a nearly inviscid flow is determined fundamentally by the surface pressure on the solid object. In the inviscid limit, the force $\mathbf{F}$ and torque $\mathbf{T}$ on a solid object with surface $\mathcal{S}$ are formulated, respectively, as

$$\mathbf{F} = -\int_{\mathcal{S}} p\hat{\mathbf{n}}ds,$$
$$\mathbf{T} = -\int_{\mathcal{S}} (\mathbf{x} - \mathbf{x}_{CM}) \times p\hat{\mathbf{n}}ds.$$

Here, $\mathbf{x}_{CM}$ is the solid object's center of mass and p is the fluid pressure. Unlike simulations based on solving the Euler equations, the vorticity transport (2) does not include a pressure term p, which is a key quantity for computing forces on a solid. Pressure can instead be computed from the surface vorticity that is emitted into the flow, which corresponds to a *vorticity flux* from the surface. First, we define this quantity and then we detail how to use it for computing fluid forces on solid objects.

6.2 Vorticity Flux

We have discussed vortex shedding as the generation of vorticity at the solid surface, and into the main flow. The rate of vorticity introduction on the main flow corresponds to a *vorticity flux* that emanates from the surface. This quantity is related to pressure changes at the solid surface and below we present it formally.

In the boundary layer, fluid motion is dominated by viscosity. Then, the vorticity transport is characterized by the following diffusion equation:

$$\frac{\partial \omega}{\partial t} = \nabla \cdot (\nu \nabla \omega).$$

Here, ν is a viscosity coefficient. The emitted vorticity from the solid object depends on the tensor $\mathbb{J} = \nu\nabla\omega$, particularly on its normal component which is the vorticity flux σ [40]:

$$\sigma = \hat{\mathbf{n}} \cdot \mathbb{J} = \nu\hat{\mathbf{n}} \cdot \nabla\omega = \nu\frac{\partial \omega}{\partial \hat{\mathbf{n}}}.$$

The vorticity flux σ can be expressed in terms of other properties. Three main contributions to σ are [27]: 1) the solid object's acceleration and body forces, 2) the surface pressure, and 3) the viscous drag. In the inviscid limit [27], σ can be rewritten as follows:

$$\begin{aligned} \sigma &= \sigma_a + \sigma_p \quad \text{with :} \\ \sigma_a &= \hat{\mathbf{n}} \times (\mathbf{a} - \mathbf{f}), \quad \sigma_p = \tfrac{1}{\rho}\hat{\mathbf{n}} \times \nabla p. \end{aligned} \tag{17}$$

Here, $\mathbf{a}$ is the solid acceleration, $\mathbf{f}$ is the acceleration due to external body forces such as gravity. Equation (17) is obtained from the vorticity flux of a general viscous flow [27].

6.3 Forces from Vorticity Flux

Computing the surface pressure from (17) is not trivial. Moreover, it is necessary to know the value of vorticity flux σ at the solid surface.

A direct way to evaluate the pressure force at the solid surface from the moments of σ_p is derived from the generalized Stokes' Theorem, which in the inviscid limit leads to [27]:

$$\begin{aligned} \mathbf{F} = -\int_{\mathcal{S}} p\hat{\mathbf{n}}ds &= -\frac{1}{2}\int_{\mathcal{S}} \mathbf{x} \times (\hat{\mathbf{n}} \times \nabla p)ds \\ &= -\frac{1}{2}\int_{\mathcal{S}} \rho\mathbf{x} \times \sigma_p \, ds. \end{aligned} \tag{18}$$

Similarly, torque on the solid due to surface pressure can be expressed in terms of the second moment of σ_p as follows:

$$\begin{aligned} \mathbf{T} &= -\int_{\mathcal{S}} (\mathbf{x} - \mathbf{x}_{CM}) \times p\hat{\mathbf{n}}ds \\ &= -\int_{\mathcal{S}} \Big(\frac{\rho}{2}\|\mathbf{x} - \mathbf{x}_{CM}\|^2\sigma_p\Big)ds. \end{aligned} \tag{19}$$

Missing still is the vorticity flux at the solid boundary. A common approximation [41], [27], [42], for vorticity flux at the no-slip solid boundary is

$$\sigma = -\frac{\gamma}{\Delta t}. \tag{20}$$

Here, γ is the vorticity of the surface vortex sheet. This approximation comes from the fact that, at each time step, the amount of vorticity introduced in the flow needs to cancel the slip velocity. This vorticity, in turn, is determined by the vortex sheet. With the approximation in (20), we rewrite the pressure contribution σ_p to the vorticity flux as follows:

$$\sigma_p = \frac{\gamma}{\Delta t} + \sigma_a, \quad \sigma_a = \hat{\mathbf{n}} \times (\mathbf{a} - \mathbf{f}). \tag{21}$$

We replace σ_p by the above expression in (18) obtaining the following definition for fluid force acting on a solid object:

$$\begin{aligned} \mathbf{F} &= -\frac{1}{2}\int_{\mathcal{S}} \rho\mathbf{x} \times \sigma_p \, ds \\ &= -\frac{1}{2}\int_{\mathcal{S}} \rho\mathbf{x} \times \Big(\frac{\gamma}{\Delta t} + \hat{\mathbf{n}} \times (\mathbf{a} - \mathbf{f})\Big) \, ds. \end{aligned} \tag{22}$$

We calculate the force and torque using the panel discretization. We evaluate the moments in (18) and (19)

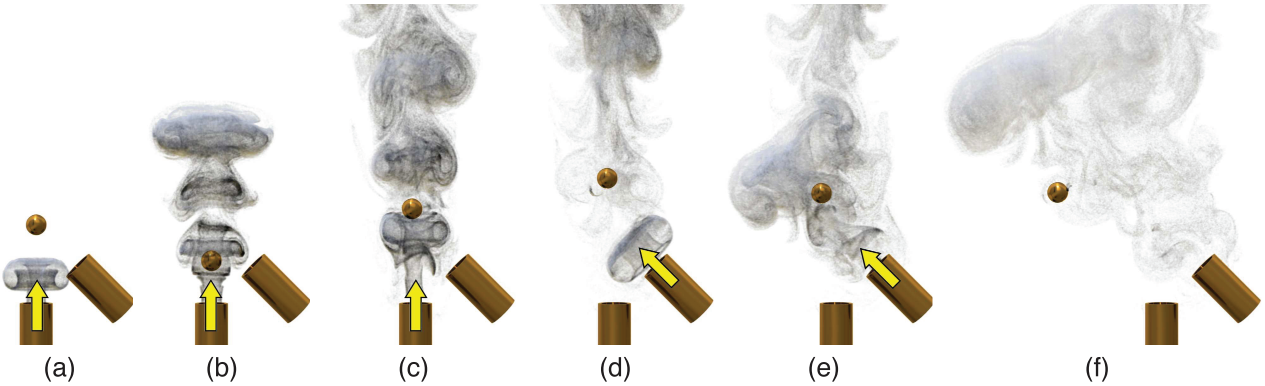

Fig. 10. Interaction between vortex jets and a unit radius sphere under the action of gravity. Here, $c_{Slip} = 0.5$ and vortex kernel radii are $\epsilon = 3.5$ for jet vortices, and $\epsilon = 1.5$ for shed vortices. (a): The sphere falls and a vertical jet is emitted. (b)-(c): The jet reaches the sphere and propels it upwards. (d) A sideways, right to left jet is emitted. (e)-(f): the second jet reaches the sphere and propels it sideways. Enabled jets are indicated with arrows.

using the control points $\mathbf{x}_j$ of each panel described in Section 4.1.3. Since we allow defining the amount of vorticity shed into the main flow using the slip coefficient c_{Slip} (Section 5.3), we scale the value of the vorticity flux at each boundary panel by c_{Slip}. Then, we compute the fluid force as follows:

$$\mathbf{F} = -\frac{1}{2}\sum_{j=1}^{N_p} \rho \mathbf{x}_j \times \left(c_{Slip} \frac{\gamma_j}{\Delta t} + \hat{\mathbf{n}}_j \times (\mathbf{a} - \mathbf{f}) \right) A_j.$$

Here, N_p corresponds to the number of panels of the solid object and A_j corresponds to the area of panel j. Using the same algebraic manipulations, we obtain the following discrete expression for torque:

$$\mathbf{T} = -\frac{1}{2}\sum_{j=1}^{N_p} \rho \left\| \mathbf{x}_j - \mathbf{x}_{CM} \right\|^2 \left(c_{Slip} \frac{\gamma_j}{\Delta t} + \hat{\mathbf{n}}_j \times (\mathbf{a} - \mathbf{f}) \right) A_j.$$

Here, $\mathbf{x}_{CM}$ is the solid object's center of mass. We compute force and torque on a solid object with these formulas and we employ these values to update its velocity and position. Fig. 10 shows an example of a sphere interacting with two vortex jets implemented with our method. Two different jets induce a change in the sphere motion as consequence of force calculations on the sphere. We summarize the different processes at each time step in the next section.

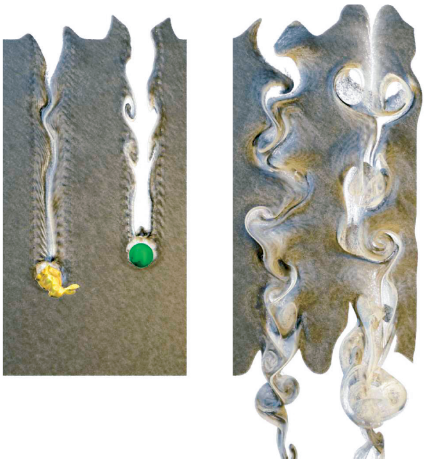

Fig. 11. Different patterns of vortex shedding produced on a sheet of smoke by a bunny model and a unit radius sphere. Shed vortices radii are set to $\epsilon = 1$.

7 Simulation Loop

A scene is composed of solids, smoke particles, and vortices. The fluid motion is determined by the vortices in the simulation and a user-defined background flow. Given the positions and velocities of the solids and particles, we perform the following operations at each time step:

- Compute the flow velocity (Section 3.2.2).
- Enforce slip boundary condition: compute the vortex sheet (Section 5.2) and vortex shedding (Section 5.3).
- Compute vortex stretching (Section 3.2.3).
- Enforce the no-penetration boundary condition (Section 4.1.3) using an updated velocity field with above shed vortices.
- Advect vortices and smoke particles using an updated velocity field using vortex mirror images.
- Compute forces on the solid object (Section 6.3) and update the solids positions.

8 Results

We show simulation results for fluid solid interplay in 3D with a sphere, a cylinder, and the Stanford bunny. To increase performance in panel calculation and vortex shedding, highly detailed surfaces can be simplified into coarser geometries. By preserving visually important features of the original mesh, flow patterns that are consistent with these features can be employed directly in rendering. In our simulations, we use 80 triangular panels for the sphere, 180 for the cylinder, and 500 for the bunny model.

The flow is induced by vortex particles and user-defined background flows. We run our experiments on a Intel Core i7-2600, 3.4-GHz CPU with 16-GB RAM and we render our scenes using Exocortex Fury 2.0.

Fig. 11 shows different vortex shedding patterns produced by a sphere and a bunny model in free fall. By employing a finite kernel radius, we limit the fluid region affected by the vortices, and vortex influence may be absent in some areas. This effect can be easily reduced by increasing the particles' radius.

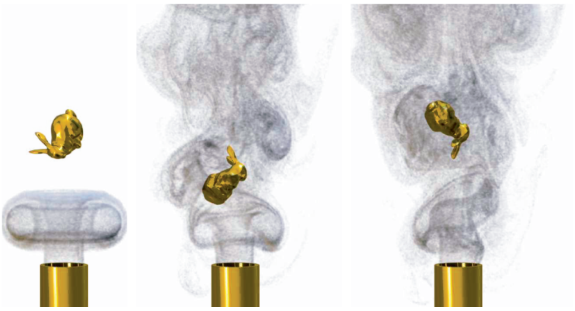

Fig. 12. Interaction between a vortex jet and a bunny model with characteristic length $L_C = 0.75$, using $c_{Slip} = 0.5$ and kernel radius $\epsilon = 0.5$ for shed vortices and $\epsilon = 3.5$ for jet vortices.

Fig. 12 shows results of interaction between a vortex jet and the bunny model. Here, the complex solid object rotates as it is lifted due to the force and torque exerted by the flow.

Fig. 13 shows an example of more complex solid fluid interaction for direct comparison with results by Klingner et al. [43]. Here, the bunny model is propelled by two sideways jets that induce the motion and rotation of the model.

Our following example mimics the interaction of rocket exhaust and a planar ground. The ground plane is large enough to prevent the flow from going around the surface. Our method produces a visually realistic simulation as shown in Fig. 14. Increased details can be obtained by adding vortices at different scales of strength and radius.

We compare our results with those of previously published vortex particle methods. Fig. 15 shows turbulence produced due to the interaction of a uniform flow and a static solid wedge. This example has been used previously to demonstrate anisotropic turbulence generated in areas of high turbulent kinetic energy [16] (Fig. 7). The approach by Pfaff et al. [16] requires simulating turbulent kinetic energy evolution along the flow and inserting noise particles corresponding to turbulence. In contrast, we do not need an energy evolution simulation to generate turbulence in the same regions. Moreover, Fig. 15 shows a clear perdiodic pattern of turbulence consistent with a von Kármán vortex street (see Fig. 6), unlike prior work by Pfaff et al. [16]. Additionally, we employ the same example to show more complex interaction, by adding a sphere which is dragged up the ramp, producing a consistent wake of turbulence.

Unlike Vortex-in-Cell (VIC) methods, we do not require an underlying mesh to perform our simulation. Our simulations are effectively unbounded as they are not limited to a grid's bounding box. Since our method does not interpolate particle data into a grid, we compare our results in terms of visual features.

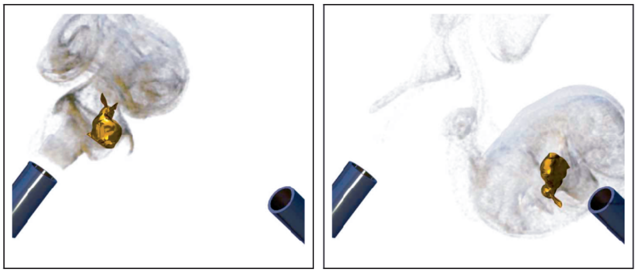

Fig. 13. Same bunny model from Fig. 12 propelled by two vortex jets from left to right and then right to left as in [43]. Here, we use $c_{Slip} = 0.3$.

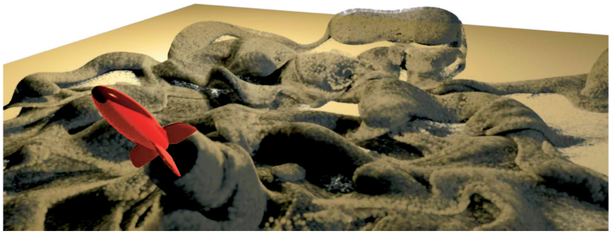

Fig. 14. Simulation of rocket exhaust, an example of a vortex jet interacting with a large planar object. The exhaust radius is 1 and shed vorticity from the surface has a radius $\epsilon = 0.5$.

Our method extends and improves results obtained using Lagrangian vortex particles in computer graphics by Park and Kim [3]. As opposed to their approach, we employ a stable vortex stretching update that does not force us to add viscosity, dampening the main flow motion to avoid numerical instability. Fig. 16 shows the result of simulating three independent smoke jets colliding, as originally proposed by Park and Kim [3]. Unlike their results, our simulation produces freely moving smoke which is not dampened by adding a high amount of viscosity to avoid instability due to vortex stretching.

Fig. 17 shows an example of different patterns in vortex shedding around a sphere, changing the slip coefficient, reproducing results previously published [3]. However, with our methods we can also realistically simulate the sphere motion induced by the flow through our force computation model, unlike any other previously published work in vortex methods for computer graphics [3], [4], [5].

Table 1 summarizes our performance results. Memory use by our method is generally negligible (less than 5 MB on average excluding marker particles). Most computational cost is due to the velocity field evaluation; hence, performance is strongly dependent on the amount of vortices in

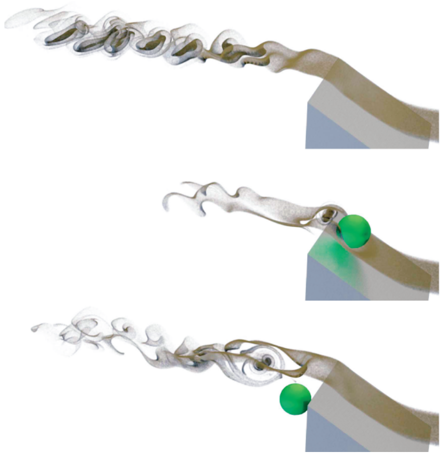

Fig. 15. Top: Simulation of uniform flow past a wedge as presented in [16]. Here, we use $c_{Slip} = 0.5$. A periodic pattern of shed vorticity forms past the solid. The shed vorticity radius is $\epsilon = 1.0$. Middle and bottom: Same scene with a unit radius sphere with $c_{Slip} = 0.5$ dragged by the flow. The sphere falls due to gravity as it reaches the top of the ramp. A consistent wake of vorticity is formed past the sphere.

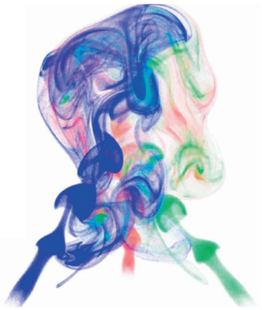

Fig. 16. Three smoke jets colliding in a scene mirroring an example from Park and Kim [3].

the simulation. A naïve solution would be to reduce the kernel radii, however, as shown in Fig. 18 this would lead to a gross visual degradation of results. As the radius decreases, the magnitude of the influence vortices on each others also decreases and, therefore, the velocity magnitude as well. We note that as opposed to previous work on hybrid simulations [16] where velocity evaluation is performed by interpolation between particles and the grid, we evaluate the field directly from particles.

Unlike other particle simulations such as SPH, vortex radii need to be large, impacting performance. A robust solution to increased performance would address the complexity of the underlying N-body problem. *Fast multipole methods* (FMM) are a large class of methods that produce controllable approximations for solving N-body problems. Our techniques can incorporate such a method, and here we outline the main aspects of such extension. Currently our system does not implement FMM, and the velocity calculations in our results were performed without such approximations.

Our velocity evaluation method can achieve a $O(n)$ performance by employing a variant of the black box FMM (bbFMM) [44]. Here, arbitrary, nonoscillatory kernels such as the one we employ can be approximated through interpolation using a truncated series of Chebyshev polynomials, which are used for fast summations on an octree structure. We describe a variation of the bbFMM [44] for our scenario in Appendix C, available in the online supplemental material. A similar approach has been previously employed [3], producing a $O(nlog(n))$ algorithm.

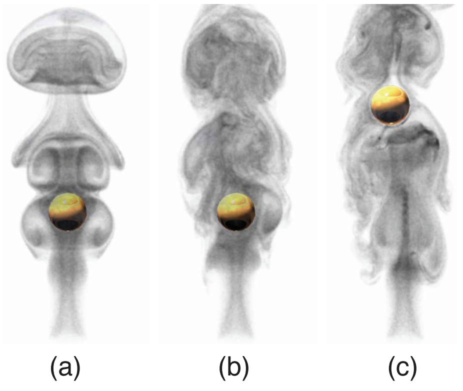

Fig. 17. Flow around a unit radius sphere: (a) using ($c_{Slip} = 0$); (b) using $c_{Slip} = 1$ and finally, (c) sphere dragged by the flow. Vortex kernel radii are $\epsilon = 1$ for shed vortices and $\epsilon = 3.5$ for jet vortices.

TABLE 1
Performance Results for 100 Frames of Simulation

Scene	Jet	Vortices Shed	Total	AVG FPS
Smoke column	480	0	480	71.5
Smoke around sphere	128	2000	2128	4.68
Smoke around cylinder	100	4988	5088	0.6904
Smoke around bunny	154	39276	39430	0.016
Bouncing sphere	906	2438	3344	1.4315
Falling objects	0	56802	56802	0.0007
Bouncing bunny	884	45530	46414	0.0114
Bouncing bunny (two jets)	640	22304	22944	0.037
Rocket exhaust	2892	1683	4576	0.84
Wedge	0	31729	31729	0.0012
Sphere and wedge	0	35719	35719	0.001

We do not include marker particles used for rendering.

Several other strategies for increasing performance may as well be employed. For instance, velocity computation can be performed on the GPU instead of CPU [5] to allow for a higher amount of parallel computations to be performed. Also, an underlying grid may be implemented to be used as a cache for flow properties computed directly from vortex particles.

9 Limitations and Future Work

Our current flow model with two-way solid fluid coupling is applicable to inviscid flows interacting with rigid objects whose volume is nonzero. This limits the applicability of our method to more general scenarios of simulating fluid interaction with deformable objects, thin shells, and filaments.

In the case of deformable objects, our methods for enforcing no-penetration can be applied directly by calculating the panel influence matrices when the solid deforms, albeit at a higher computational cost. However, calculating the solid's deformation due to the flow requires computing the fluid pressure at each of the solid's surface elements.

In the case of thin shells, which are often represented as an open surface, source panel methods are not an adequate solution as the velocity that cancels the flow on one side of the surface, adds up on the other side, which would lead to simulation artifacts. Hence, the need for a different model for enforcing no-penetration. In the case of filaments, if they are defined by a series of connected control points, these can be easily advected to follow the flow; however, an

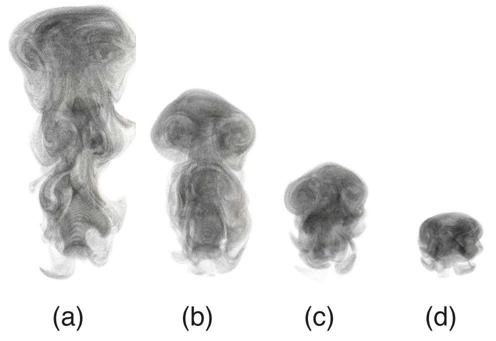

Fig. 18. Smoke simulation using different kernel radius ϵ after 300 iterations. (a): $\epsilon = 2.5$; (b): $\epsilon = 2.0$; (c): $\epsilon = 1.5$; (d): $\epsilon = 1.0$. Smoke's emission radius is 1.0.

adequate shedding method should be devised to model a slip condition.

There are also a number of useful directions for further research. One of them is the application of our methods to viscous flows. Here, the assumption that surface vorticity degenerates into a vortex sheet is no longer valid [40]. Surface vorticity must then be found by explicitly solving a Poisson equation on the solid surface [45]. Also, extending the interaction of vorticity to general surfaces would enable the application of our methods to simulate free-surface flows.

10 Conclusions

Our method introduces a novel approach to model the interplay between solids and fluids using Lagrangian vortex methods. We introduce a novel technique to stably solve the vortex stretching term in the vorticity transport equation, which produces improved results compared to previously published simulations using Lagrangian vortex particle methods.

We introduce a novel method to computer graphics for enforcing no-penetration boundary conditions on rigid solids by solving a boundary integral problem imposing conditions only on the normal components of the flow. We model the boundary layer evolution through shedding vorticity at solid surfaces and advecting this vorticity in the flow obtaining physically consistent results. Together this enables the visually plausible simulation of turbulent flows.

We compute the pressure distribution on the solid surface from the vorticity flux enabling us to calculate the pressure force exerted by the fluid on the solid. This permits two-way coupling of multiple solids with the fluid enforcing the appropriate boundary conditions.

Acknowledgments

The authors would like to thank NSERC, Mitacs Canada and FedDev Ontario for their support of this work. They would also like to thank Erik Uggeldahl from Exocortex Technologies, Inc. for his assistance in rendering, and to Christopher Batty, UBC, for his comments on an early version of this work.

References

[1] R. Fedkiw, J. Stam, and H.W. Jensen, "Visual Simulation of Smoke," *Proc. ACM SIGGRAPH '01,* pp. 15-22, 2001.

[2] T. Pfaff, N. Thuerey, A. Selle, and M. Gross, "Synthetic Turbulence Using Artificial Boundary Layers," *ACM Trans. Graphics,* vol. 28, pp. 121:1-121:10, Dec. 2009.

[3] S.I. Park and M.J. Kim, "Vortex Fluid for Gaseous Phenomena," *Proc. ACM SIGGRAPH/Eurographics Symp. ComputerAnimation (SCA),* pp. 261-270, 2005.

[4] A. Angelidis, F. Neyret, K. Singh, and D. Nowrouzezahrai, "A Controllable, Fast and Stable Basis for Vortex Based Smoke Simulation," *Proc. ACM SIGGRAPH/Eurographics Symp. Computer-Animation (SCA),* pp. 25-32, 2006.

[5] S. Weißmann and U. Pinkall, "Filament-Based Smoke with Vortex Shedding and Variational Reconnection," *ACM Trans. Graphics,* vol. 29, pp. 115:1-115:12, July 2010.

[6] A.M. Kuethe and C.Y. Chow, *Foundations of Aerodynamics: Bases of Aerodynamics Design.* John Wiley and Sons, 1998.

[7] L. Prandtl, "On Motion of Fluids with Very Little Viscosity," *Proc. Third Congress Math.,* 1904.

[8] M.J. Lighthill, "Introduction: Boundary Layer Theory," *Laminar Boundary Theory,* L. Rosenhead, ed., pp. 46-113, Oxford Univ. Press, 1963.

[9] N. Foster and D. Metaxas, "Realistic Animation of Liquids," *Graphical Models Image Processing,* vol. 58, pp. 471-483, Sept. 1996.

[10] J. Stam, "Stable Fluids," *Proc. ACM SIGGRAPH '99,* pp. 121-128, 1999.

[11] M. Desbrun and M.-P. Cani, "Smoothed Particles: A New Paradigm for Animating Highly Deformable Bodies," *Proc. Eurographics Workshop Computer Animation and Simulation (EGCAS '96),* R. Boulic and G. Hegron, eds., pp. 61-76, Aug. 1996.

[12] M. Müller, D. Charypar, and M. Gross, "Particle-Based Fluid Simulation for Interactive Applications," *Proc. ACM SIGGRAPH/ Eurographics Symp. ComputerAnimation (SCA),* pp. 154-159, 2003.

[13] M.N. Gamito, P.F. Lopes, and M.R. Gomes, "Two-Dimensional Simulation of Gaseous Phenomena Using Vortex Particles," *Proc. EG Workshop Computer Animation and Simulation,* pp. 3-15, 1995.

[14] A. Selle, N. Rasmussen, and R. Fedkiw, "A Vortex Particle Method for Smoke, Water and Explosions," *Proc. ACM SIGGRAPH 2005 Papers,* pp. 910-914, 2005.

[15] J.-C. Yoon, H.R. Kam, J.-M. Hong, S.J. Kang, and C.-H. Kim, "Procedural Synthesis Using Vortex Particle Method for Fluid Simulation," *Computer Graphics Forum,* vol. 28, no. 7, pp. 1853-1859, 2009.

[16] T. Pfaff, N. Thuerey, J. Cohen, S. Tariq, and M. Gross, "Scalable Fluid Simulation Using Anisotropic Turbulence Particles," *Proc. ACM SIGGRAPH Asia 2010 Papers,* pp. 174:1-174:8, 2010.

[17] A. Golas, R. Narain, J. Sewall, P. Krajcevski, P. Dubey, and M. Lin, "Large-Scale Fluid Simulation Using Velocity-Vorticity Domain Decomposition," *ACM Trans. Graphics,* vol. 31, no. 6, pp. 148:1-148:9, Nov. 2012.

[18] T. Pfaff, N. Thuerey, and M. Gross, "Lagrangian Vortex Sheets for Animating Fluids," *ACM Trans. Graphics,* vol. 31, no. 4, pp. 112:1-112:8, July 2012.

[19] D. Kim, S.W. Lee, O. young Song, and H.-S. Ko, "Baroclinic Turbulence with Varying Density and Temperature," *IEEE Trans. Visualization and Computer Graphics,* vol. 18, no. 9, pp. 1488-1495, Sept. 2012.

[20] F. Sadlo, R. Peikert, and M. Sick, "Visualization Tools for Vorticity Transport Analysis in Incompressible Flow," *IEEE Trans. Visualization and Computer Graphics,* vol. 12, no. 5, pp. 949-956, Sept. 2006.

[21] C. Batty, F. Bertails, and R. Bridson, "A Fast Variational Framework for Accurate Solid-Fluid Coupling," *ACM Trans. Graphics,* vol. 26, article 100, July 2007.

[22] A. Robinson-Mosher, T. Shinar, J. Gretarsson, J. Su, and R. Fedkiw, "Two-Way Coupling of Fluids to Rigid and Deformable Solids and Shells," *ACM Trans. Graphics,* vol. 27, pp. 46:1-46:9, Aug. 2008.

[23] M. Müller, S. Schirm, M. Teschner, B. Heidelberger, and M. Gross, "Interaction of Fluids with Deformable Solids: Research Articles," *Computer Animation and Virtual Worlds,* vol. 15, pp. 159-171, July 2004.

[24] N. Akinci, M. Ihmsen, G. Akinci, B. Solenthaler, and M. Teschner, "Versatile Rigid-Fluid Coupling for Incompressible SPH," *ACM Trans. Graphics,* vol. 31, no. 4, article 62, 2012.

[25] J.H. Walther and A. Larssen, "Two Dimensional Discrete Vortex Method for Application to Bluff Body Aerodynamics," *J. Wind Eng. and Industrial Aerodynamics,* vol. 67-68, pp. 183-193, 1997.

[26] D. Russell and Z.J. Wang, "A Cartesian Grid Method for Modeling Multiple Moving Objects in 2d Incompressible Viscous Flow," *J. Computational Physics,* vol. 191, pp. 177-205, Oct. 2003.

[27] J. Wu and J. Wu, "Vorticity Dynamics on Boundaries," *Advances in Applied Mechanics,* J.W. Hutchinson and T.Y. Wu, eds., vol. 32, pp. 119-275, Elsevier, 1996.

[28] J. Strickland, V. Porter, G. Homicz, and A. Gossler, "A Vortex Code for Flow over Rigid or Flexible Bluff Bodies*," *J. Turbulence,* vol. 3, no. 1, 2002.

[29] Q. Li, H. Wu, M. Guo, and J.-Z. Wu, "Vorticity Dynamics in Axial Compressor Flow Diagnosis and Design—Part ii: Methodology and Application of Boundary Vorticity Flux," *J. Fluids Eng.,* vol. 132, no. 1, p. 011102, 2010.

[30] M. Coquerelle and G.H. Cottet, "A Vortex Level Set Method for the Two-Way Coupling of an Incompressible Fluid with Colliding Rigid Bodies," *J. Computational Physics,* vol. 227, pp. 9121-9137, Nov. 2008.

[31] J.C. Wu, "Theory for Aerodynamic Force and Moment in Viscous Flows," *AIAA J.,* vol. 19, pp. 432-441, Apr. 1981.

[32] B.N. Shashikanth, A. Sheshmani, S.D. Kelly, and J.E. Marsden, "Hamiltonian Structure for a Neutrally Buoyant Rigid Body Interacting with n Vortex Rings of Arbitrary Shape: The Case of Arbitrary Smooth Body Shape," *Theoretical and Computational Fluid Dynamics,* vol. 22, no. 1, pp. 37-64, 2008.

[33] P. Ploumhans, G.S. Winckelmans, J.K. Salmon, A. Leonard, and M.S. Warren, "Vortex Methods for Direct Numerical Simulation of Three-Dimensional Bluff Body Flows: Application to the Sphere at re = 300, 500, and 1000," *J. Computational Physics,* vol. 178, pp. 427-463, May 2002.

[34] A. Angelidis and F. Neyret, "Simulation of Smoke Based on Vortex Filament Primitives," *Proc. ACM SIGGRAPH/Eurographics Symp. Computer Animation (SCA),* pp. 87-96, 2005.

[35] R. Krasny and L. Kaganovskiy, "Computation of Vortex Ring Dynamics," *Proc. Int'l Conf. High Reynolds Number Vortex Interactions,* pp. 46-113, 2005.

[36] J.L. Hess, "Higher Order Numerical Solution of the Integral Equation for the Two-Dimensional Neumann Problem," *Computer Methods in Applied Mechanics and Eng.*, vol. 2, no. 1, pp. 1-15, 1973.

[37] R. Bridson, *Fluid Simulation for Computer Graphics.* A.K. Peters, 2008.

[38] P.G. Saffman, *Vortex Dynamics.* Cambridge Univ. Press, 1992.

[39] A.E. Perry, M.S. Chong, and T.T. Lim, "The Vortex-Shedding Process Behind Two-Dimensional Bluff Bodies," *J. Fluid Mechanics,* vol. 116, pp. 77-90, Mar. 1982.

[40] H.G. Cottet and P. Koumoutsakos, *Vortex Methods: Theory and Practice.* Cambridge Univ. Press, 2000.

[41] C. Casciola, R. Piva, and P. Bassanini, "Vorticity Generation on a Flat Surface in 3d Flows," *J. Computational Physics,* vol. 129, no. 2, pp. 345-356, 1996.

[42] J.D. Eldredge, "Numerical Simulation of the Fluid Dynamics of 2d Rigid Body Motion with the Vortex Particle Method," *J. Computational Physics,* vol. 221, pp. 626-648, Feb. 2007.

[43] B.M. Klingner, B.E. Feldman, N. Chentanez, and J.F. O'Brien, "Fluid Animation with Dynamic Meshes," *ACM Trans. Graphics,* vol. 25, no. 3, pp. 820-825, July 2006.

[44] W. Fong and E. Darve, "The Black-Box Fast Multipole Method," *J. Computational Physics,* vol. 228, no. 23, pp. 8712-8725, Dec. 2009.

[45] D. Rempfer, "On Boundary Conditions for Incompressible Navier-Stokes Problems," *Applied Mechanics Rev.*, vol. 59, no. 3, pp. 107-125, May 2006.

Mauricio Vines received the bachelor's degree in engineering sciences from the Catholic University of Chile, in 2002, where he graduated as a computer engineer in 2004. He has been working toward the PhD degree at the University of Ottawa since 2007. He has developed fluid-simulation systems for interactive haptic applications and high-resolution simulations systems aimed at the VFX industry. His research interests include the simulation of natural phenomena for computer graphics applications and specifically the simulation of fluid phenomena.

Ben Houston is the founder and CEO of Ottawa-based Exocortex Technologies, Inc. Exocortex creates software solutions for the VFX and game industry. Exocortexs current client list includes dozens of the top VFX firms around the world. Exocortex software was recently used on Harry Potter and the Deathly Hallows Part 2 (2011), Titanic 3D (2012), The Avengers (2012), Snow White and the Huntsman (2012), and The Amazing Spider-Man (2012.) He has designed, lead, and/or managed development of seven commercially successful software products for the VFX industry as well as many special projects. Software he has created has grossed over $10M in sales in the global VFX industry over the last decade. Author of eight published papers and three US patents with more than 150 citations. Academic work has been reproduced by both Pixar, Industrial Light and Magic, Stanford University, UCLA, UBC, and CalTech.

Jochen Lang received the MSc degree in computer science from York University, Toronto, Canada, and the PhD degree in computer science from the University of British Columbia, Canada, in 2001. From 2002 to 2004, he was a postdoctoral researcher at the Max-Planck-Institut für Informatik, Saarbrücken, Germany. Currently, he is an associate professor with the School of Electrical Engineering and Computer Science, University of Ottawa, Canada, where he is a member of the Distributed and Collaborative Virtual Environments Research Laboratory (DISCOVER) and the VIVA laboratory. His research interest is focused on physics-based and data driven modeling for computer graphics and computer haptics. He is working on image-based models, computational photography, and measurement-based 3D model acquisition. He is a member of the IEEE.

Won-Sook Lee received the BSc degree in mathematics from POSTECH, South Korea, the MScs degrees in mathematics from POSTECH and in systems science from the National University of Singapore (Singapore) and Diplôme d'Etudes Supérieures en Systèmes d'Information from the University of Genève (Switzerland). She received the PhD degree in human modeling for animation from the University of Genève. She has working experience in several industries such as Korea Telecom, Samsung (South Korea) as well as Eyematics Inc. (US). She has been with the School of Electrical Engineering and Computer Science, University of Ottawa, Canada since 2003 where she is currently an associate professor. Her research interests include computer graphics and computer animation. She has worked on virtual human modeling and animation, face recognition, music analysis, haptic, medical imaging, and fluid. She is a member of several conferences, Journals, and NSERC evaluation committees.