TwisterForge: controllable and efficient animation of virtual tornadoes

Jiong Chen
Inria & École Polytechnique, CNRS (LIX)
IP Paris, France
jiong.chen@inria.fr

James Gain
University of Cape Town
Cape Town, South Africa
jgain@cs.uct.ac.za

Jean-Marc Chomaz
Ecole Polytechnique, CNRS (LadHyX)
IP Paris, France
jean-marc.chomaz@polytechnique.edu

Marie-Paule Cani
École Polytechnique, CNRS (LIX)
IP Paris, France
marie-paule.cani@polytechnique.edu

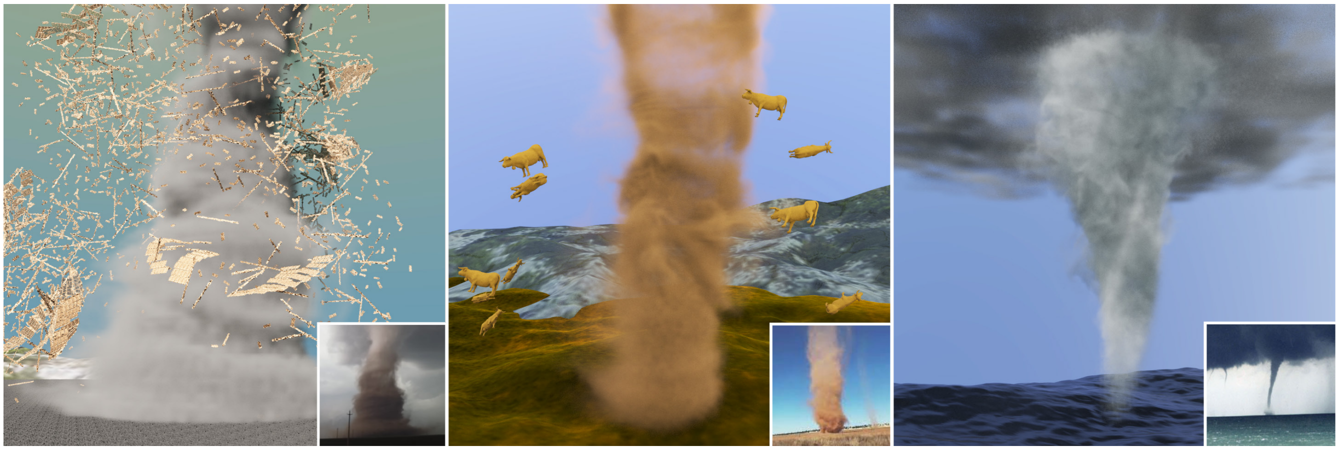


Figure 1: Virtual Tornadoes — Our method allows artists to intuitively control the visual appearance of tornadoes and animate their movement and deformation, supporting interaction with solid objects and rough terrain. Based on a physically inspired core-and-funnel model, our approach balances animation efficiency, control flexibility and visual plausibility (see inset reference photos of real tornadoes).

Abstract

We propose a simple method for the intuitive authoring and efficient animation of virtual tornadoes. Users control the tornado kinematics by sketching two types of curves to specify the initial geometry of the tornado's core and the profile of the surrounding swirling air, known as the funnel. The first input, a 3D curve, initializes the core as a vortex filament. This filament induces a *swirl* flow and advects according to its initial curvature, resulting in progressive bending and twisting. The second input consists of one or multiple 2D profile curves that parameterize the Stokes stream function, governing the *radial* and *axial* motion of the air around the core and thereby dictate the funnel shape over time. The core and funnel profile are coupled in local frames through closed-form velocities, which together describe the rotation, sliding and uplift within the tornado's air volume. As shown in our case studies, our method provides a controllable and efficient way to animate visually plausible tornadoes capable of tearing off infrastructure and transporting debris, as well as interacting with uneven terrain.

CCS Concepts

• **Computing methodologies** → *Shape modeling*.

Keywords

Animation of Natural Phenomena, Vortex filament, Stokes Stream Function, Sketch-based Modeling.

ACM Reference Format:
Jiong Chen, James Gain, Jean-Marc Chomaz, and Marie-Paule Cani. 2024. TwisterForge: controllable and efficient animation of virtual tornadoes . In *The 16th ACM SIGGRAPH Conference on Motion, Interaction, and Games (MIG '24), November 21–23, 2024, Arlington, VA, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3677388.3696335

1 Introduction

Tornadoes, already among the most destructive of weather phenomena, are becoming more frequent and intense due to climate change [Tippett et al. 2016]. Given their destructive capacity, physics and meteorology have appropriately focused on high-fidelity but

MIG '24, November 21–23, 2024, Arlington, VA, USA

complicated and computationally costly simulation of these phenomena. At the same time, although their appearance is both visually striking and emotionally charged, there has only been limited research into the controllable reproduction of such vortex-like phenomena in Computer Graphics. In addition to cinema and gaming, such controllable yet visually plausible animation models would be useful in education, museography and other virtual environments, such as training simulators.

A vortex occurs in nature as a rapidly rotating column of air that may contain dust, debris, condensed water vapour, entrained water or even flames, depending on the circumstances. The physics of vortex formation and sustenance are complex and vary with vortex type. Given amenable conditions in a supercell thunderstorm, a tornado may form when rotating air near and parallel to the ground, arising from convective gusts and eddies of warm and cool air, is forced vertical by a clash with a rotating downdraft. Subsequently, the column is dragged and stretched by the thunderstorms movement above and its tethering to the ground below. The resulting tornado varies in diameter from about 20 to 200m at the base, with an upper limit of 4km, and in strength from a barely visible funnel to an elemental force capable of shearing buildings and sending cars flying. Apart from tornadoes, other vortex phenomena include funnels (where the vortex does not reach the ground), water spouts (where water from seas or lakes is entrained), gustnadoes (shallow and weak ground-level vortices) and land spouts (cylindrical vortices formed along cold fronts rather than thunderstorms). Dust devils, steam whirls, and fire whirls are weaker effects mostly caused by localized heating and distinguished by the type of material incorporated. Any advances in the modeling of tornadoes would, with small modification, be applicable to capturing these effects as well.

Previous approaches to tornado simulation tend to fall into one of two extremes: either simplified analytic models, which use regression fitting to match their parameters to measured behaviour, or full numerical solutions, which simulate Navier-Stokes dynamics on an Eulerian grid. The former is generally too simplified to describe the complex vortex structures and motions of tornadoes, while the latter are computationally expensive and difficult to configure for desired effects. Neither of these approaches adequately addresses the issue of user control for flexible animations.

In contrast, our aim is to develop an intuitive authoring tool dedicated to the modeling and efficient animation of visually plausible tornadoes. To achieve this we adopt a middle ground approach in the form of a two-layer *core-and-funnel* model, which the user initializes by sketching two types of curves: one for the 3D core, and a second, consisting of one or multiple 2D curves, for prescribing the profile of the surrounding air volume over the time. The kinematics of the tornado core is then captured by modeling it as a vortex filament, subject to self-advection through the Biot-Savart law. The swirl flow induced by this filament is extended to the surrounding air volume, augmented with radial and axial velocities that are derived from the *Stokes stream functions* defined through our 2D profiles. Through this decoupling we avoid any need for grid-based discretization, iterations of boundary configuration, and costly numerical solves of general-purpose fluid simulation, greatly facilitating the flexibility and direct control in the animation workflow.

In addition to easing the design of varied vortex types and funnel shapes, this model is able to reproduce complex effects evident in nature, such as tornadoes with a variety of flexible cores that sweep up debris, follow ground topography, and advance with the storm front. Our new model, illustrated in Fig. 1, thus supports simple but expressive authoring, while also achieving a balance between computational efficiency and visual realism.

2 Previous Work

The structure and dynamics of tornadoes have been intensively studied in meteorology, from both a theoretical [Davies-Jones et al. 2001; Klemp 1987; Lewellen 1993; Rotunno 2013] and experimental [Church et al. 1979; Ward 1972] perspective. Two broad strategies have emerged for modeling and simulating tornado dynamics: solving Navier-Stokes equations for the general case and deriving closed-form solutions for specific vortex structures. Unfortunately, as will become clear, neither of these approaches is ideal as a basis for *efficient* and *controllable* authoring and animation of visually realistic tornadoes.

Numerical simulations: These methods target the reproduction of high-resolution, fully three-dimensional unsteady tornado vortex dynamics, in particular the turbulent flows that arise from interaction with the ground and debris. In this regard, Large Eddy Simulation (LES) [Deardorff 1970] represents a particularly effective choice. The core idea is to address the Navier-Stokes equation at different scales: large-scale motion is solved directly, while unresolved, small-scale turbulence is accounted for by a sub-grid model. In the context of tornado simulation, LES has been used to model the atmospheric boundary layer and near-surface tornado dynamics [Lewellen et al. 2000, 1997], as well as debris transport using a two-fluid model [Lewellen et al. 2008] that distinguishes the near-ground debris cloud from the more elongated carrier flow. Such two-fluid models for tornado simulation have also been explored in computer graphics [Herrera et al. 2024; Liu et al. 2007, 2008], supported by efficient GPU implementations.

Despite achieving high fidelity and physical realism, numerical simulations frequently encounter significant challenges in animation control. One such challenge arises from the intricate setup of tornado dynamics, including initial and boundary conditions, turbulence models, force terms, multi-fluid interaction, and thermodynamics, all of which influence the shape and evolution of a tornado through a highly nonlinear process and demand strong expertise to configure them properly for targeted scenarios. Moreover, full-space simulations are computationally expensive, which could hinder rapid feedback for user interaction and thus raise costs of iterative design.

Analytic vortices: Compared to solving the full-space Navier-Stokes equations, the theory of vortex dynamics provides a terser description of swirling flow [Alekseenko et al. 2007; Saffman 1993]. This class of methods enables vortex-specific solutions to the Navier-Stokes or Euler equations by assuming helical symmetry [Varaksin 2017; Varaksin and Ryzhkov 2023], easing the numerical difficulties associated with a full 3D solve. Unfortunately, these simplifying assumptions fail to account for certain emergent structures, such as the four characterizing regions identified by Lewellen [Lewellen 1993], particularly in the case of closed-form expressions. While

Shtern et al. [1997] go part way to solving this by deriving a family of explicit vortex solutions for swirling flows, their solutions are limited to rational polynomials expressed in cylindrical coordinates relative to a straight rectilinear core.

In summary, although the simplicity, attendant computational efficiency and parametric control of such methods is attractive, they often lack generality in terms of handling more complex cores, varied vortex structures, and ground interaction. With the intent of offering more intuitive control while balancing fidelity and computational cost, we build on these analytic solutions and their spline-parameterized extension by replacing the linear tornado core with a more general vortex filament representation.

Vortex filaments are curves in a fluid that compactly represent regions of concentrated vorticity. They are widely used in physics as a dynamic model for both conventional and super fluids [Koplik and Levine 1993]. In computer graphics, they have been used to improve the efficiency and control of smoke simulation [Angelidis et al. 2006; Chern et al. 2016; Weißmann and Pinkall 2010] and to model underwater bubble rings [Padilla et al. 2019] as well as stellar corona [Padilla et al. 2019]. While ideal for capturing the vorticity concentrated at the heart of a tornado, the attendant field dynamics remain computationally expensive.

To overcome the limitations of existing analytic and numerical models and to strike the necessary balance between visual plausibility, controllability, and efficiency, we represent the tornado core as a single, self-advected vortex filament, subject to the Biot-Savart law [Rosenhead 1930]. We spread its motion to the surrounding air volume using a parametric Stokes stream function [Shtern et al. 1997], thereby extending analytical vortex solutions to encompass more varied structures, which offers useful control over the tornado's shape and appearance.

3 Overview

3.1 A Primer on Tornadoes

It is worthwhile providing a capsule summary of the physics of tornado initiation, structure, and dynamics that inspire our two-layer core-and-funnel model.

The prevailing theory of tornado formation is that warm, moist air near the ground meets cooler, drier air aloft, creating instability in the atmosphere and finally leading to a thunderstorm. Within the thunderstorm, strong updrafts of air develop, tilting a nearly horizontal cylinder of rotating air upward into a helix-like vertical structure, which is further intensified and stretched from the ground to a portion of the cloud base that extends downward.

After initiation, a tornado grows until maturity and can then remain self-sustaining for a period lasting from a few minutes to around an hour. A mature tornado typically has a multi-vortex structure, which can be divided into four flow regions (see Fig. 2), each with a distinct underlying physical mechanism [Lewellen 1977].

The central region (1a) is the *core* where vorticity is concentrated with strong rotation. In this region, the effect of buoyancy and viscosity is negligible compared to pressure and angular momentum. As a result, the dynamics can be assumed to be barotropic and inviscid, and are adequately modeled by a thin vortex tube.

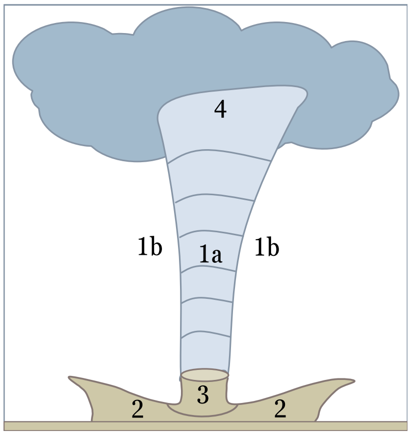


Figure 2: Tornado structure: A generic tornado can be conceptually divided into four critical regions [Lewellen 1977]: the core (1a), outer flow (1b), a near-ground boundary layer (2), a corner region (3), and a top region (4).

The core's surroundings have more complex mechanics. The *outer flow* region (1b) extends from the core to encompasses the volume of approaching air. Here, the horizontal pressure gradient balances the centrifugal force of the swirling core. Due to the decrease in pressure, air will condensate from water vapour into moisture and thus form the visible funnel cloud. As for the *boundary layer* (2), the friction with the ground reduces tangential wind velocity near the surface, leading to an imbalance of pressure gradient and centrifugal force, which consequently drives a strong inflow at the bottom. The boundary layer connects to the *corner* region (3), where instability such as turbulence or vortex breakdown may occur to dissipate the core vorticity due to strong viscosity. Finally, in the *top* region (4), the tornado is embedded in its parent storm under buoyancy forces. The actual mechanism is complex, heavily depending on the mesoscopic storm, and very little information is known about this region.

Typically, the visual appearance of a tornado is the joint effect of flows in these regions, composed of water droplets condensed out due to low pressure as air is drawn down from the cloud base into the condensation funnel and dust and debris is drawn up from

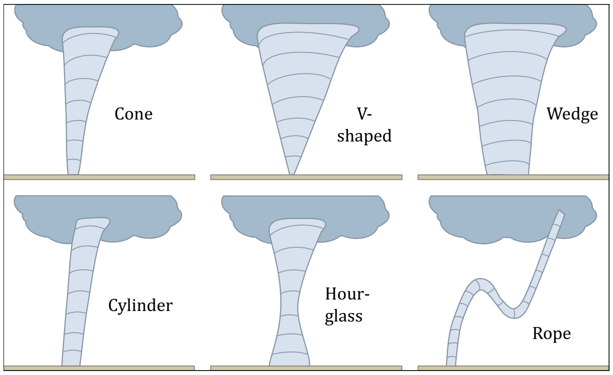


Figure 3: The variety of tornado funnel shapes.

the ground into the debris cloud. The presence and extent of either or both of these will vary with the ambient conditions and may even involve different materials, such as fire, sea water, and steam, in the case of other vortex phenomena. In particular, the shape and diameter of the funnel may vary widely, encompassing cones, v-shapes, wedges, cylinders, hour-glasses, and ropes (see Fig. 3). Its evolving form depends on many factors, such as the local advection and convection dynamics of the prevailing storm, the height of the wall cloud, the humidity and temperature of the air, and fine debris on the ground, to name just a few.

3.2 Framework

Given the complexity and uncertainty of tornado physics, our goal is to instead leverage the efficiency of analytical methods to provide both authoring control over the core shape and funnel profiles, and visual realism in terms of overall motion, vortex strength, and interaction with movable obstacles and the terrain. Specifically, users can easily choose whether the core is deforming rapidly or slowly, and whether the surrounding funnel of the condensed air maintains a constant profile or evolves its profile over time, based on their authoring requirements.

In practice, users provide two types of curves: a 3D spline $\boldsymbol{q}(s)$ to represent the initial shape of the filament core as Fig. 4 depicts (any piecewise parametric curve affording an arc-length parametrization, such as a Bézier curve, can be used), and a 2D spline $r = f(z)$ to define the side-on half profile of the condensation funnel around the core. Users can optionally add extra 2D splines $r = f_{t_i}(z)$ to encode keyframe profiles at different time-steps t_i. This provides an effective means of controlling the funnel shape, as a generalized cylinder surrounding the core. This profile can then either remain fixed or be interpolated over time.

We then build the kinematics of the tornado based on our core-and-funnel representation: the filament core that drives vorticity and the volumetric sheath that extends this flow to the surrounding space: In Section 4, the kinematics of the tornado core is modeled by a vortex filament that obeys the Biot-Savart law, incorporating self-induction and additional no-penetration boundary conditions for anchoring to the terrain. In Section 5, we model the sheath of axisymmetric swirling flow around the core by exploiting a spline-based parameterization of the Stokes stream function. This allows us to derive a consistent velocity field around the core capable of portraying the transport of particles and solid bodies. In Section 6, we explore the extent of control through various case studies and experiments that demonstrate the range of achievable tornadoes in terms of shape control, debris clouds as well as terrain interaction.

4 Tornado Core

The first layer of our model initializes the tornado core from a smooth curve. During subsequent animation we model the core as a vortex filament — a spatial curve with attached local rotary flow. This enables us to capture both its self-advected motion and the local vorticity it induces.

Although vortex filaments have already been suggested for modeling the motion of a tornado core [Aiki and Iguchi 2012], a number of questions remain unresolved, including how to incorporate user control, handle numerical instabilities that arise on the filament curve, satisfy boundary conditions at the base of the filament, and achieve the discretization necessary for efficient evaluation.

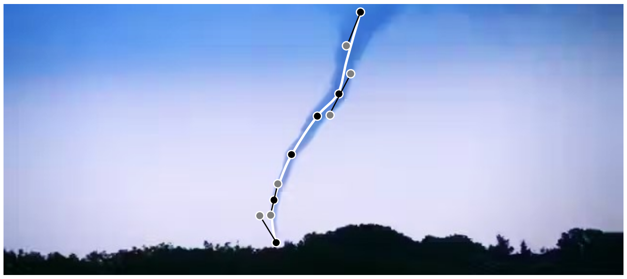
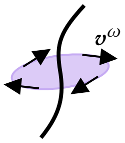


Figure 4: Core initialization: The user can specify the geometry of the core by placing and editing spline control points, e.g., fitting the spatial orientation of an existing tornado.



Core kinematics. In our method, vorticity is generated by the core and entrains the surrounding air. Specifically, given the arc-length curve $\boldsymbol{q}(s) : [0, L] \rightarrow \mathbb{R}^3$, the swirling velocity $\boldsymbol{v}^{\omega}(\boldsymbol{p})$ at a point $\boldsymbol{p}$ near the curve can be computed by the well-known Biot-Savart law in its integral formulation [Saffman 1993]. A direct application of this law leads to divergence in the velocity evaluation as $\boldsymbol{p}$ approaches $\boldsymbol{q}(s)$ [Batchelor 2000], so instead we adopt the Rosenhead-Moore approximation [Rosenhead 1930]. This adds a regularization term by demarcating a filament of constant vorticity surrounding the core with radius a. The flow integral under this approximation is:

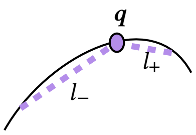
$$\boldsymbol{v}^{\omega}(\boldsymbol{p}) = -\frac{\Gamma}{4\pi}\int_0^L \frac{(\boldsymbol{p} - \boldsymbol{q}(s)) \times \partial_s \boldsymbol{q}}{(\|\boldsymbol{p} - \boldsymbol{q}(s)\|^2 + \mu^2)^{\frac{3}{2}}}\,\mathrm{d}s, \qquad (1)$$

where Γ is a circulation term that measures vortex strength, and $\mu^2 = e^{-3/2}a^2$ is the regularization term that prevents the singularity as $\|\boldsymbol{p} - \boldsymbol{q}(s)\| \rightarrow 0$ and produces asymptotically correct speed for a thin vortex ring with a core of constant vorticity [Leonard 1985]. This allows the core in its dynamic form $\boldsymbol{q}(s, t)$ to be driven by self-induced velocity:

$$\frac{\partial \boldsymbol{q}}{\partial t} = \boldsymbol{v}^{\omega}(\boldsymbol{q}) = -\frac{\Gamma}{4\pi}\int_0^L \frac{[\boldsymbol{q}(s,t) - \boldsymbol{q}(s',t)] \times \partial_{s'}\boldsymbol{q}}{(\|\boldsymbol{q}(s) - \boldsymbol{q}(s')\|)^2 + \mu^2)^{\frac{3}{2}}}\mathrm{d}s'. \qquad (2)$$

The local induction approximation. When approximating a smooth curve with piecewise linear segments, the contribution of the cross product in Eq. (2) to a point is zero on its incident segments [Padilla et al. 2019]. To compensate, the vanishing integral can be approximated by the local induction approximation (LIA) [Hama 1962], which assumes that the filament core is so thin that its motion is almost entirely governed by the local curvature. Thus, the velocity driving the kinetics of the tornado is composed of LIA ($\boldsymbol{v}^{LIA}(\boldsymbol{q})$) and vorticity ($\boldsymbol{v}^{\omega}(\boldsymbol{q})$) components:

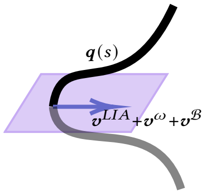
$$\begin{aligned} \frac{\partial \boldsymbol{q}}{\partial t} &= \boldsymbol{v}^{LIA}(\boldsymbol{q}) + \boldsymbol{v}^{\omega}(\boldsymbol{q}) = \frac{\Gamma}{4\pi}\left(\frac{\partial \boldsymbol{q}}{\partial s} \times \frac{\partial^2 \boldsymbol{q}}{\partial s^2}\right)\log\left(\frac{2\sqrt{l_- l_+}}{e^{1/4}a}\right) \\ &- \frac{\Gamma}{4\pi}\int_0^L \frac{(\boldsymbol{q}(s) - \boldsymbol{q}(s')) \times \partial_{s'}\boldsymbol{q}}{(\|\boldsymbol{q}(s) - \boldsymbol{q}(s')\|^2 + \mu^2)^{\frac{3}{2}}}\,\mathrm{d}s', \end{aligned} \qquad (3)$$



where l_- and l_+ designate the length of segments incident on the point $\boldsymbol{q}$. The velocity arising from the LIA lies along the binormal of the curve, and its magnitude is proportional to the local curvature. Intuitively, the filament will deform, or bend, faster where this curvature is greater. In addition to LIA and vorticity, a global velocity term $\boldsymbol{v}^{global}$ can be also introduced into Eq. (3) to account for environmental effects, such as a constant wind velocity that moves the tornado to match the motion of its parent storm, as shown in in Figs. 8 and 11.

No-penetration boundary conditions. Real tornadoes are not always anchored to the ground, but, when they are, their interaction with the topography of the terrain should be considered.

Accurate modeling of boundary layers must consider the friction between the tornado and the ground, and properly address the boundary conditions. For instance, swirling motion typically adheres to no-slip conditions and radial motion to no-stress conditions [Davies-Jones 2008]. Because the boundary layer is very thin, computationally expensive high-resolution simulations are required for properly imposing these conditions. However, for our lightweight animation, it is sufficient to focus on the basic immutable boundary effects, such that *the swirling flow never enters the ground.* For a planar surface with normal $\boldsymbol{n}$, this boundary condition amounts to enforcing $\boldsymbol{n}^\top \dot{\boldsymbol{q}}(s)|_{s=0} = 0$, thus ensuring that the normal velocity vanishes. Inspired by Schwarz [1985], a simple way to impose this constraint in a filament-based representation is to add a correction velocity $\boldsymbol{v}^{\mathcal{B}}$ to $\boldsymbol{v}^{LIA}+\boldsymbol{v}^{\omega}$, which is computed from a vortex filament *mirrored* with respect to the surface (see inset). While this choice oversimplifies the interaction between the core and the surface, typical boundary layer effects, such as strong jet flows near the surface, can be reproduced in other ways, such as through an appropriate funnel profile. In practice, we represent the terrain as a height field, so that, with suitable interpolation, the tangent plane is smoothly varying and can be easily computed at each time step for the velocity correction. Fig. 5 illustrates the effectiveness of the boundary effects achieved through reflected filaments.



Discretization. The 3D curve $\boldsymbol{q}(s)$ provided by the user as an initial condition is first discretized into a set of line segments $\{(\boldsymbol{q}_j, \boldsymbol{q}_{j+1})\}_{j=1}^{E}$. While the core will wind and spin according to the tornado dynamics, this initial curve does provide a strong indicator of the evolving shape.

At each timestep, $\boldsymbol{v}^{\omega}$ at vertex $\boldsymbol{q}_i$ is evaluated as a piece-wise integration of the Rosenhead-Moore core over all segments, yielding:

$$\boldsymbol{v}^{\omega}(\boldsymbol{q}_i) = -\frac{\Gamma}{4\pi}\sum_{j=1}^{E}\int_0^{L_j} \frac{(\boldsymbol{q}_i - \boldsymbol{q}_j)\times(\boldsymbol{q}_{j+1}-\boldsymbol{q}_j)/L_j}{(\|\boldsymbol{q}_i - \boldsymbol{q}_j - s(\boldsymbol{q}_{j+1}-\boldsymbol{q}_j)/L_j\|^2 + \mu^2)^{\frac{3}{2}}}\mathrm{d}s, \tag{4}$$

where $L_j = \|\boldsymbol{q}_{j+1} - \boldsymbol{q}_j\|$. We note that the above integral can be evaluated as a closed-form expression [Weißmann and Pinkall 2010]. For the local induction approximation, the cut-off lengths l_- and l_+ are simply chosen as the lengths of the two line segments incident on $\boldsymbol{q}_i$, compensating for the vanishing integration that occurs in Eq. (4). The binormal term in LIA is approximated as [Padilla et al. 2019]:

$$\frac{\partial \boldsymbol{q}}{\partial s}\times\frac{\partial^2 \boldsymbol{q}}{\partial s^2}\bigg|_{\boldsymbol{q}=\boldsymbol{q}_i} \approx \frac{2\boldsymbol{e}_{i-1}\times\boldsymbol{e}_i}{\|\boldsymbol{e}_{i-1}\|\|\boldsymbol{e}_i\|(\|\boldsymbol{e}_{i-1}\| + \|\boldsymbol{e}_i\|)}, \tag{5}$$

where $\boldsymbol{e}_i = \boldsymbol{q}_{i+1} - \boldsymbol{q}_i$.

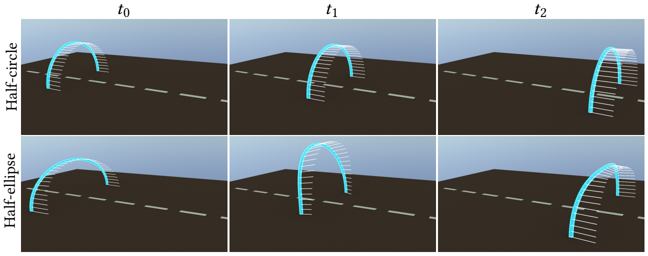


Figure 5: No-penetration boundary validation: For filaments anchored at both ends to the plane, the (invisible) reflected portion will create a closed curve and enforce the no-penetration conditions. The velocity (white lines emerging from the curve) is driven by curvature. The half-circle (top row) merely translates from left to right along the dashed line, because the velocities are uniform, while the half-ellipse (bottom row) also bends and flexes as a result of differences in velocity magnitude. As a validation, the results match real world experiments[1].

5 Water Condensation Funnel

While our vortex filament model adequately describes the kinematics of the tornado core, it is insufficient for the full structure of a tornado, which extends well beyond the bounds of a thin vortex tube. In particular, the Biot-Savart law does not include any *axial* or *radial* flow, which is needed for the upward motion of air and embedded debris within the outer flow regions and near-ground boundary layers (see Fig. 2), and the emergence of different funnel shapes (see Fig. 3). This necessitates the augmentation of our model.

We base this augmentation on the spline profile-curve(s) provided by the user (see Section 3), which describes the vortex that encapsulates the core. This not only provides an intuitive mechanism for controlling the geometry of the air volume, but also leads to a closed-form vortex velocity field for efficient animation. To achieve this, we exploit parameterized *Stokes stream functions* using smooth C^2 splines to describe axisymmetric vortex structures.

In the rest of this section, we first explain our solution in a rectilinear coordinate system, and then adapt it to a curved configuration. Note that in our approach, the profile describing the condensation funnel around the core can either remain constant or be interpolated between keyframes, facilitating flexible control over its temporal behaviour.

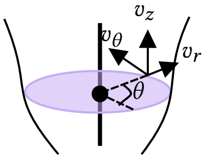
5.1 Some Background on Axisymmetric Vortices

While overly simplified models for axisymmetric vortices [Alekseenko et al. 2007] are not able to capture complex swirling flows, Shtern et al. [1997] describe a more general family of axisymmetric,

[1] A video showing the dynamics of half vortex rings under water is found at https://www.youtube.com/watch?v=72LWr7BU8Ao.

viscous swirling vortices encompassing jets and tornadoes. Their closed-form solutions are derived from the Navier-Stokes equation in cylindrical coordinates, with the assumption that the radial velocity and swirl are decoupled from the axial velocity, and the vertical pressure gradient is non-zero. This yields:

$$\begin{cases} v_r = \mu \, \mathrm{Re}/r, \\ v_\theta = \mu \Gamma / r, \\ v_z = \mu \left[W_c + W_p r^2 + W_r r^{\mathrm{Re}} \right]. \end{cases} \tag{6}$$



Here, v_r, v_θ and v_z are the radial velocity, azimuthal (swirl) velocity and axial velocity, respectively (see inset). Constants include the kinematic viscosity (μ), the radial Reynolds number (Re), and circulation (Γ). The axial velocity is further characterized by uniform (W_c) and non-uniform shear (W_p, W_r) components.

Due to axial and radial flow, adjusting these constants can be used to vary the funnel shape. Mathematically speaking, these shapes are axisymmetric algebraic surfaces parameterized by the level set of the associated Stokes stream function, which reads as:

$$\Psi(r,z) = \frac{W_c \mu}{2} r^2 + \frac{W_p \mu}{4} r^4 + \frac{W_r \mu}{2 + \mathrm{Re}} r^{\mathrm{Re}+2} - \mathrm{Re} \, \mu z = 0. \tag{7}$$

Depending on the sign of Re, the velocity in Eq. (6) either gives rise to a vortex sink (swirling top-down) with a negative Re, or a vortex source (swirling bottom-up) with a positive Re.

5.2 A Spline-driven Stokes Stream Function

Shtern vortices are driven by viscosity and, while this adequately captures the corner flows and boundary layers of a tornado's structure (see Fig. 2), it does not suffice for the outer and top flows that transport and stretch the vorticity. Geometrically, their approach is capable of parameterizing simple shapes, such as cones and wedges, but it still does not permit variability sufficient to capture all cases (see Fig. 3). For instance, it is not possible to describe an hourglass funnel in such a formulation, because the height z is simply a polynomial function of the radial distance r according to Eq. (7). Recognizing this, Shtern et al. [1997] had to introduce two sets of parameters for tornadoes: one for the strong jet near the ground and another for the widening bulge at higher altitudes, separately.

Instead, we propose a more flexible and controllable approach to modeling general vortices. Rather than deriving each velocity component from the Navier-Stokes equation, we directly parameterize and manipulate the Stokes stream function $\Psi(r,z)$, so that the radial and axial velocities become:

$$\begin{cases} v_r = -\dfrac{1}{r} \dfrac{\partial \Psi}{\partial z}, \\ v_z = +\dfrac{1}{r} \dfrac{\partial \Psi}{\partial r}. \end{cases} \tag{8}$$

On the one hand, this technical choice allows for intuitive and flexible control over the funnel shape surrounding the core. On the other hand, the continuity equation for axisymmetric, incompressible flow is automatically satisfied by construction, so that we always have:

$$\frac{1}{r} \frac{\partial (r v_r)}{\partial r} + \frac{\partial v_z}{\partial z} = 0, \tag{9}$$

which ensures that the mass of air is conserved within the swirling flow. This property has an intuitive influences on air transport: thinner sections will lift air upwards faster than broader sections.

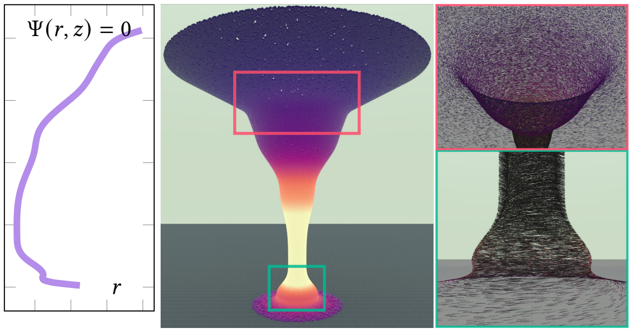


Figure 6: Spline-based vortex. Left: the sketched spline curve representing the Stokes stream function. Middle: particles are advected by our spline-based vortex with the velocity magnitude color-coded from low (dark purple) to high (light yellow). The velocity of particles increase as they approach the core. Right: close-ups of particle velocity at the bottom and top of the vortex.

The crucial step for control is to employ a user-specified cubic spline curve $r = f(z)$, representing the side-on half profile of the tornado (see Section 3), to define the zero level set of the stream function $\Psi(r,z)$. The choice to parameterize in r ($z = f(r)$) rather than z ($z = f(r)$) allows for non-monotonic variation in radius with height, and hence support for tornadoes with a characteristic "C"-shaped profile (see Fig. 6). More general yet would be to introduce two splines, $r = g(\gamma)$ and $z = h(\gamma)$, parameterized according to a parameter γ, but this would significantly complicate both the user interface and the derivation of $\Psi(r,z)$.

In practice, an initial placement of spline control points $\{r_m, z_m\}_m$ is generated according to Eq. (6). This provides a starting point for further editing, which can continue until a desired tornado profile is obtained.

Since the zero level set is being edited, the stream function itself becomes:

$$\Psi(r,z) = \alpha \left(r - f(z) \right), \tag{10}$$

with the constant α used to re-scale the magnitude of the radial and axial velocities. Once Eq. (10) is differentiated in r and z, the radial v_r and axial velocities v_z can be obtained directly from Eq. (8).

The next step is to couple the surrounding vortex with the tornado core. We begin by assuming a rectilinear core and then later generalize this to a full 3D curve in Section 5.4. Since the Stokes stream function only returns radial and axial velocities, we can simply constrain the missing azimuthal (swirl) velocity v_θ to accord with the Biot-Savart law locally, such that $v_\theta = \|\boldsymbol{v}^\omega\|$. As a result, our first-pass spline-based funnel model takes the form:

$$\begin{cases} v_r = \dfrac{\alpha}{r} \nabla_z f, \\ v_\theta = \dfrac{\Gamma}{2\pi r}, \\ v_z = \dfrac{\alpha}{r}. \end{cases} \tag{11}$$

Notice that these three velocities all contain singularities as r approaches zero. To over come this, we regularize the distance following the same idea as Eq. (1) by adding a constant μ^2.

This closed-form analytic vortex structure avoids the need for an iterative numerical solution and importantly also bypasses the tedious work of recreating the ambient conditions necessary in a fully numerical tornado simulation. However, this explicit specification comes at a cost: the resulting vortex does not generally satisfy the Navier-Stokes equation, as demonstrated in Appendix A.

The vortex velocity for a point in cylindrical coordinate space provided by Eq. (11) can now be used to advect particles and apply linear and torque forces to rigid bodies for visualization purposes. Furthermore, by directly editing the cubic splines for the core and funnel, users are able to achieve the full range of funnel shapes shown in Fig. 3 (see also the accompanying video).

5.3 Keyframe Interpolation of Stream Functions

One significant advantage of representing the funnel geometry using 2D parametric curves is that it allows for direct blending between keyframes. If a user provides multiple keyframe profiles, our model can interpolate between frames rather than being restricted to a steady funnel around the core. This provides users with the capability to approximate the evolution of a tornado over time.

Once the user has sketched a sequence of profiles $f_{t_i}(z)$ representing the funnel configurations at keyframes $\{t_i\}_i$, the intermediate profile at any time t between $[t_i, t_{i+1}]$ can then be smoothly interpolated using Catmull-Rom splines [Catmull and Rom 1974]:

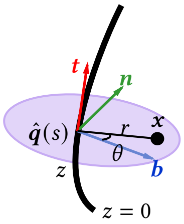
$$f(z,t) = \frac{1}{2}\begin{bmatrix}1 & w & w^2 & w^3\end{bmatrix}\begin{bmatrix}0 & 2 & 0 & 0\\ -1 & 0 & 1 & 0\\ 2 & -5 & 4 & -1\\ -1 & 3 & -3 & 1\end{bmatrix}\begin{bmatrix}f_{t_{i-1}}(z)\\ f_{t_i}(z)\\ f_{t_{i+1}}(z)\\ f_{t_{i+2}}(z)\end{bmatrix}, \quad (12)$$

where $t = (1-w)t_i + wt_{i+1}$ with $w \in [0,1]$. If $f_{t_{i-1}}$ (*resp.* $f_{t_{i+2}}$) does not exist, it is simply set to f_{t_i} (*resp.* $f_{t_{i+1}}$). Figs. 9 and 10 present examples of keyframing the funnel profiles to animate the expansion of a tornado and its life cycle, from inception to dissipation, respectively.

5.4 Adapting to a Curved Core

While the spline-based funnel model as presented (Eq. (11)) supports the animation of tornadoes with various funnel shapes, it is currently defined for the restrictive case of a rectilinear core. To permit the general case, we align the vortex funnel with a local frame-based parameterization of the curved core. Specifically, a given point in the funnel is expressed in the coordinate frame of the closest point on the core, enabling velocities to adapt to the curved configuration.

Suppose that a smooth frame function $\boldsymbol{R}(s)$ is defined over the core, with tangent vector $\boldsymbol{t}$, normal vector $\boldsymbol{n}$ and binormal vector $\boldsymbol{b}$ (see inset). For a nearby point in space ($\boldsymbol{x} \in \mathbb{R}^3$), we can compute its local cylindrical coordinates (r, θ, z) with respect to the frame of the closest point on the curve, denoted as $\hat{\boldsymbol{q}}(s)$. In such a case, the radius r is the distance between $\boldsymbol{x}$ and $\hat{\boldsymbol{q}}(s)$, the azimuth angle θ is formed between $\boldsymbol{b}$ and $\boldsymbol{x}-\hat{\boldsymbol{q}}(s)$, and the height z is the arc length at $\hat{\boldsymbol{q}}(s)$.



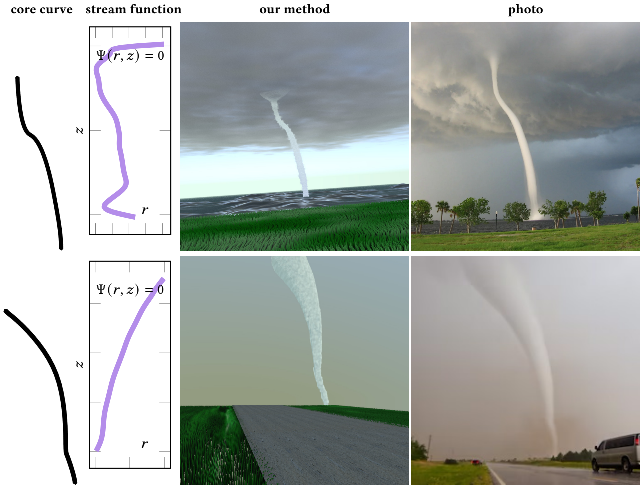


Figure 7: Static cores: Some tornadoes are relatively mild, deforming slowly over time, especially when seen in the distance. By disabling vortex filament dynamics while retaining the stream functions, our method supports the virtual reproduction of tornadoes (bottom) and waterspouts (top).

After evaluating velocities v_r, v_θ and v_z using Eq. (11) and the frame relative r and z, the velocity vector in Euclidean space is recovered by:

$$\boldsymbol{v}(\boldsymbol{x}) = \big(\boldsymbol{b}(s)\ \boldsymbol{n}(s)\ \boldsymbol{t}(s)\big)\begin{pmatrix}\cos(\theta)v_r - \sin(\theta)v_\theta\\ \sin(\theta)v_r + \cos(\theta)v_\theta\\ v_z\end{pmatrix}. \quad (13)$$

As before, at each timestep the core evolves according to Eq. (3), but this is now followed by an update of the alignment of local frames. Note that, since our vortex model is axisymmetric, velocities are independent of θ and hence invariant to the choice of normal and binormal vectors. In practice, we simply compute the Bishop frame for each line segment [Bergou et al. 2008]. To improve smoothness, we compute a local frame for each point $\boldsymbol{q}_i$ by averaging its two neighboring segment frames, and finally represent $\boldsymbol{R}(s)$ as a piecewise linear function, inspired by Phong deformation [James 2020]. The actual interpolation of $\boldsymbol{R}$ is performed in logarithmic space, where each rotation matrix is mapped to an anti-symmetric matrix and the final frames are recovered via the exponential map.

6 Results

In this section, we provide implementation and performance details and demonstrate the flexibility and generality offered by our framework in terms of authoring and animating virtual tornadoes.

6.1 Implementation Details

Our methods are implemented in C++, CUDA, and SideFX Houdini 19. For testing purposes we execute on a laptop, containing an AMD Ryzen 7 5800H CPU, with 8 cores and 16GB RAM, as well as an NVIDIA Geforce RTX 3060 Laptop GPU with 6GB RAM.

Computing tornado kinematics. For each timestep, integration is performed in two stages: (a) self-advection of the filament curve, and (b) particle transport within the funnel.

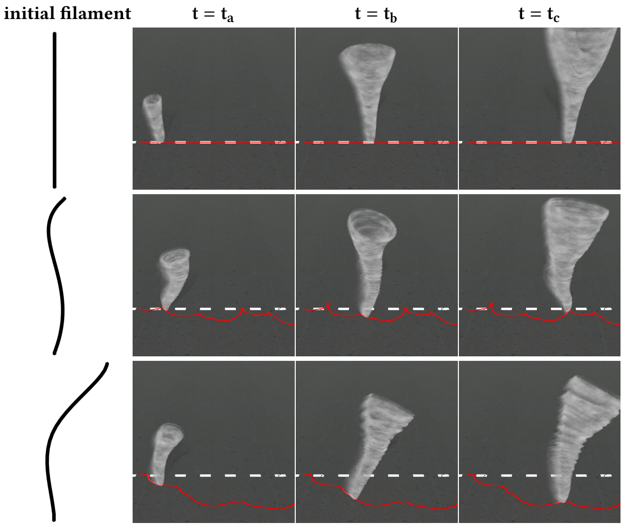


Figure 8: Dynamic cores: Differences in filament initialization affect the core dynamics. A straight filament does not bend or twist over time and traces a correspondingly straight path along the ground (in red). In contrast, a curved filaments flexes more dynamically as reflected by the trajectory of its tip.

In the first stage, the time integration of the filament curve kinematics from Eq. (3) is performed on the CPU using the 4^{th} order explicit Runge-Kutta method with a step duration of 10^{-3}s for all our examples. The core radius a is set to 0.05% of the filament length. In some cases, such as Figs. 8 to 12, the tornado is also globally translated by an environmental velocity v^{global}.

In the second stage, the kinematics of particles in the air volume around the core are calculated on the GPU. These represent the dust and moisture that makes a tornado funnel visible. At each timestep, we generate new particles randomly either at the base of the tornado or surrounding the filament curve. These are then advected along with existing particles according to Eq. (11), using a 2^{nd} order Runge-Kutta integration scheme. For each particle, we leverage the GPU to compute its local cylindrical coordinates and then return its velocity, as described in Section 5.4.

Rendering. Once the simulation has run its course, we export the per-frame position, velocity, and stream function value for all particles, both on the core and in the air volume. This exported data is loaded into Houdini for volume rendering of the final animation. Realism can be further enhanced by adding turbulence using Houdini's Pyro smoke solver, such as in Figs. 9 to 12.

While we chose to perform rendering as a post-process, there is nothing that fundamentally prevents per-frame rendering, as might be required for interactive applications, such as games.

One-way interaction with solid bodies. Tornadoes often cause severe damage, tearing at infrastructure and dragging along objects in their path, including vehicles, animals and people. Due to this devastating power, the reverse feedback of small-scale solids on tornado dynamics can be considered negligible over a short time scale. Accordingly, drag forces and associated torques are exerted by the tornado on the solid objects, propelling their motion. This rigid body interaction is implemented using the Bullet Physics subsystem in Houdini. For instance, to animate the disintegration of the houses in Fig. 11 we model their roofs and walls out of individual planks with virtual spring connections, which stretch and break under the extreme forces induced by the tornado.

6.2 User Control

A significant advantage of our approach is that it provides an intuitive and controllable interface for artists based on curves. User control takes three forms: a 3D curve to initialize the core, 2D profile curves to delineate the funnel, and optional provision of a terrain to serve as an anchoring boundary condition.

Control over the core. The kinematics of the filament core can run in two modes: without or with self-advection. The former is a useful simplification for low-energy vortices, particularly when viewed from afar over the short term, as illustrated by our visual replication of real tornadoes in Fig. 7. In contrast, the latter capture the whip and flail exhibited by high-energy tornadoes, particularly in their embryonic phase. As a case in point, Fig. 8 shows how different filament initializations change the dynamics of both a tornado's geometry and trajectory.

Control over the funnel profile. The funnel cloud surrounding the core is critical to the evolving tornado shape. While direct authoring of the zero level set of the Stokes stream function ignores much of the inherent complexity, it supports both visual plausibility and user control, as demonstrated by Figs. 7, 9 and 11 in which the 2D profile maps dynamically to the axisymmetric funnel shape.

Terrain interaction. It has been established that the behaviour and life cycle of tornadoes is heavily impacted by topography. For instance, tornadoes are reported to occur less frequently over rugged terrain [Gaffin and Parker 2006]. Unfortunately, the factors governing the formation, near-ground vortex structure, and funnel trajectory for a given terrain are too complex to model precisely and predict reliably and remain an open topic in both numerical simulation [Satrio et al. 2020] and practical case studies [Schneider 2009]. Our workaround is to mirror the filament about the surface tangent plane to ensure that flow is tangential. Despite its simplicity, this boundary condition nevertheless supports mass conservation. For instance, Fig. 12 shows an example of a tornado traversing rough terrain with the tornado base noticeably tilting to align, and Fig. 9 shows similar alignment even when the surface alters dynamically.

6.3 Performance

We have measured the per-step time cost for simulating a tornado core (Eq. (3)) and its surrounding air volume by particle transport (Eq. (11)). In our timing experiments we seed the tornado base with 10 new particles on each timestep. The cost of particle advection is linear and quickly grows to dominate the core, which typically has less than 100 vertices (see Fig. 13). In total, thanks to our GPU implementation, the per frame cost is below 30ms for up to 1.5×10^5 particles.

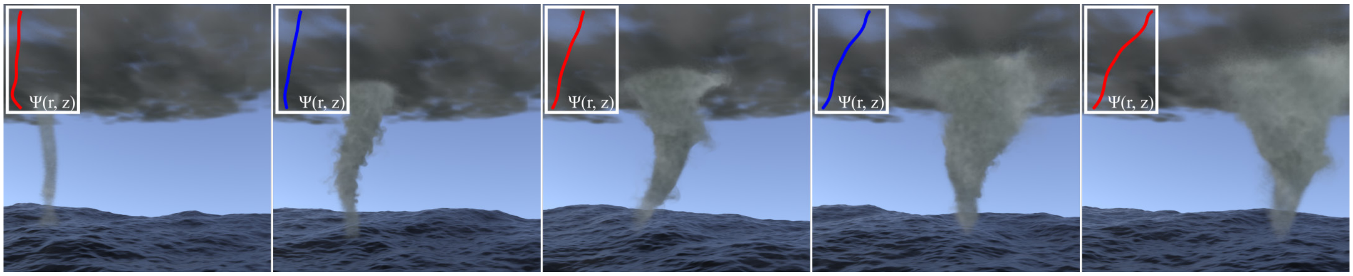


Figure 9: Control over the funnel profile: The waterspout moves across the sea with its funnel progressively expanding. This process is controlled by prescribed keyframe profiles of the Stokes stream function (red curves), which are interpolated over time for smooth animation (blue curves).

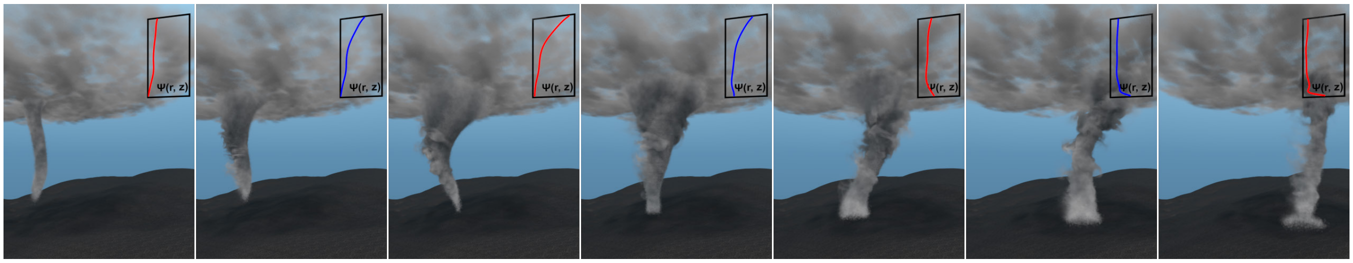


Figure 10: Life cycle: By prescribing and interpolating the profile curves of Stokes stream functions our method is capable of plausibly replicating the life cycle of a tornado from inception to dissipation.

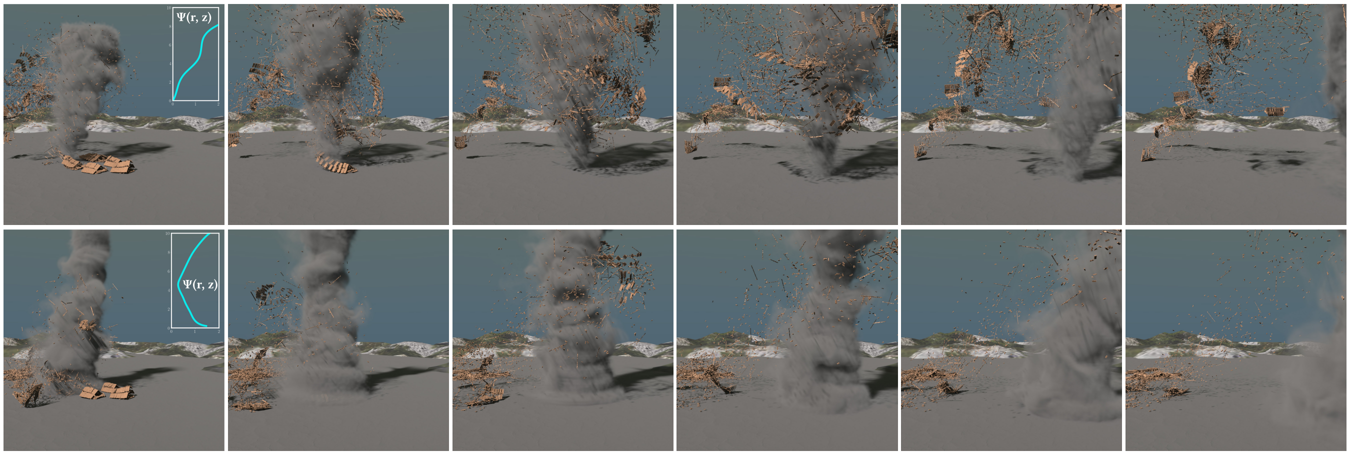


Figure 11: Uni-directional interaction with solids: Drag forces cause the disintegration of dwellings and transport of wooden debris for two different funnels: a wedge with distinct steps (top) and an hour-glass (bottom).

The optional procedural turbulence process runs via the Houdini smoke solver, and depends on the voxel size used for rasterization. Since we do not require fine-scale turbulence, we achieve interactive rates of around 10 frames per second for most of the cases.

6.4 Extensions

Our method is primarily focused on tornadoes. Nevertheless, the central idea of decomposing a vortex structure into a filament core and surrounding funnel to circumvent costly general-purpose simulation is applicable to other swirling-flow phenomena.

As an early demonstration of generality, we have animated a fire whirl with an hour-glass funnel, as depicted in Fig. 14. We believe our sketch-based approach has the potential to enhance the production of similar animations, thus offering a convenient, computationally efficient, and controllable alternative for visual effects.

6.5 Limitations and Future Work

While our core-and-funnel model is inspired by tornado physics, it lacks complete physical accuracy due to simplifications made in the interests of computational efficiency and authoring control.

First, our spline-based approach to modeling tornado vortex structure significantly simplifies the region-dependent physics. In particular, the core and its surrounding vortex are only loosely

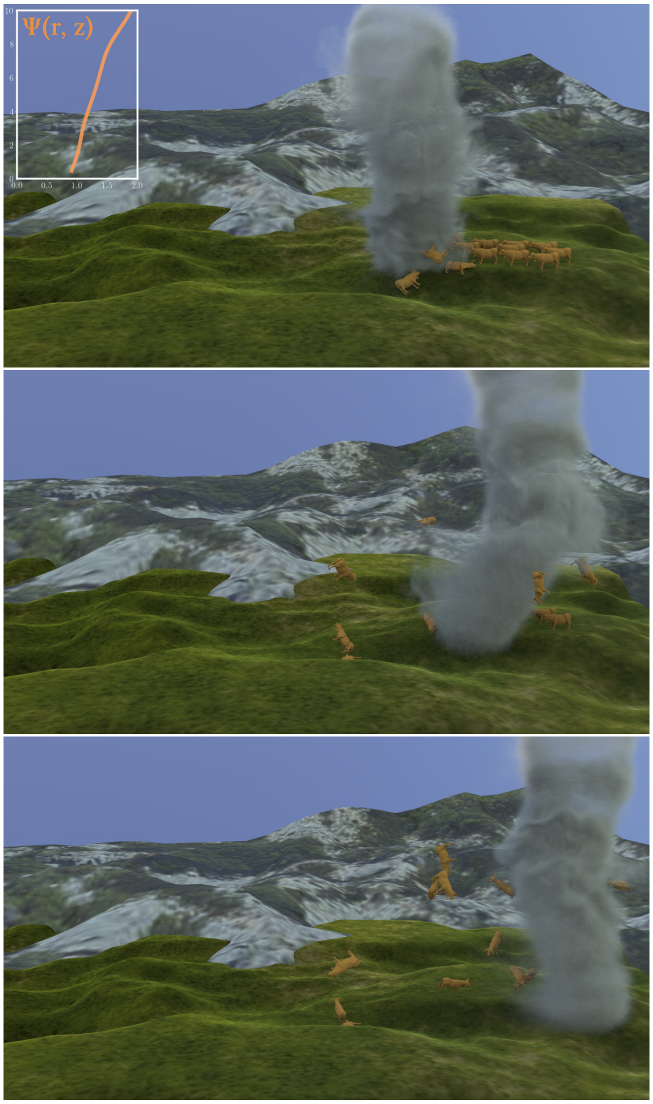


Figure 12: Rough terrain: The no-penetration boundary condition imposed by the reflected filament keeps the tornado nearly orthogonal even when the terrain is fairly uneven.

coupled through Eq. (11) by matching the swirling velocity. In reality, the swirling air volume should actively influence the core dynamics, transferring momentum from the surrounding vortex. In our setup the funnel adapts to the core but not vice versa.

Second, the actual interaction between a tornado and terrain surface is far more complex than what can be achieved by directly enforcing no-penetration conditions for basic mass conservation. An accurate boundary layer model should also account for friction forces and model terrain roughness with a high-resolution discretization.

Third, our model also does not take into consideration two-way interaction between tornadoes and their encompassing environment, including the dynamics of environmental temperature, pressure, wind, moisture, and cloud structure.

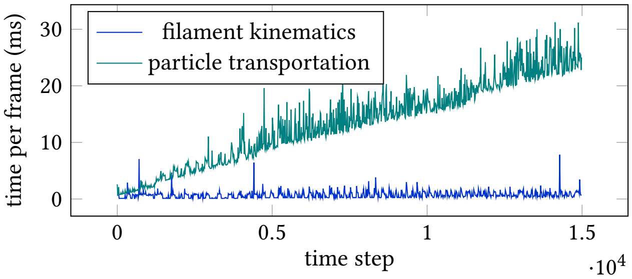


Figure 13: Timing: The time cost for evolving the filament curve (blue) and transporting air particles (teal) is recorded per time step. The overhead for core kinematics is negligible compared to particle advection. Here, the performance is linear because we seed 10 new air particles on each timestep, and the overall cost is directly proportional to the number of particles. The per frame computation time remains below 30ms even for the $1.5{\times}10^5$ particles present in the final frames.

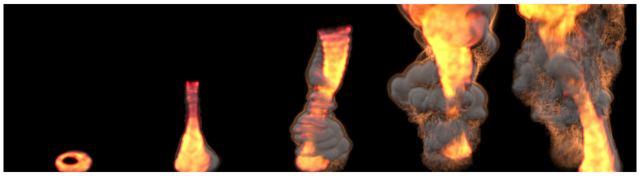


Figure 14: Fire whirl: Using the same profile as in Fig. 11 (bottom), we animated a fire whirl in Houdini by initializing per-particle temperature and flame intensity via Perlin noise. The smoke is generated in burning regions and lifted by buoyancy.

These limitations are the necessary trade-offs made in exchange for controllability and efficiency. To improve realism while maintaining control would likely require integrating stream function authoring into a full Navier-Stokes simulation. This strategy would necessitate both computationally expensive space-time optimization and precise modeling of the environment in different regions of the tornado.

7 Conclusion

In this work, we have proposed a sketch-based authoring method coupled with a simplified, two-layer model for the visual animation of a tornado. This enables a user to flexibly manipulate the visual shape and dynamics of a tornado by sketching two types of curves, one for the core and another for the extent of the entrained air volume. These curves are used to direct the kinematic motion of a vortex filament and surrounding axisymmetric flow, respectively. While this represents a simplification of the actual dynamics, our framework strikes a balance between efficiency, user control and visual plausibility which can benefit applications like special effects and video games.

Acknowledgments

This work was funded by the Uber chair at Ecole Polytechnique, the National Research Foundation of South Africa (Grant Number:129257) and the Hi!Paris fellowship 'Creative AI'.

References

David J Acheson. 1991. Elementary fluid dynamics.

Masashi Aiki and Tatsuo Iguchi. 2012. Motion of a vortex filament in the half-space. *Nonlinear Analysis: Theory, Methods & Applications* 75, 13 (2012), 5180–5185.

Sergeĭ Vladimirovich Alekseenko, Pavel Anatolevich Kuibin, and Valeriĭ Leonidovich Okulov. 2007. *Theory of concentrated vortices: an introduction.* Springer Science & Business Media.

Alexis Angelidis, Fabrice Neyret, Karan Singh, and Derek Nowrouzezahrai. 2006. A Controllable, Fast and Stable Basis for Vortex Based Smoke Simulation. In *ACM-SIGGRAPH/EG Symposium on Computer Animation (SCA 06).* Vienne, Austria, pages 25–32.

G. K. Batchelor. 2000. *An Introduction to Fluid Dynamics.* Cambridge University Press. https://doi.org/10.1017/CBO9780511800955

Miklós Bergou, Max Wardetzky, Stephen Robinson, Basile Audoly, and Eitan Grinspun. 2008. Discrete elastic rods. In *ACM SIGGRAPH 2008 papers.* 1–12.

Edwin Catmull and Raphael Rom. 1974. A class of local interpolating splines. In *Computer aided geometric design.* Elsevier, 317–326.

Albert Chern, Felix Knöppel, Ulrich Pinkall, Peter Schröder, and Steffen Weißmann. 2016. Schrödinger's smoke. *ACM Transactions on Graphics (TOG)* 35, 4 (2016), 1–13.

CRo Church, JT Snow, GL Baker, and EM Agee. 1979. Characteristics of tornado-like vortices as a function of swirl ratio: A laboratory investigation. *Journal of the Atmospheric Sciences* 36, 9 (1979), 1755–1776.

Robert Davies-Jones. 2008. Can a descending rain curtain in a supercell instigate tornadogenesis barotropically? *Journal of the Atmospheric Sciences* 65, 8 (2008), 2469–2497.

Robert Davies-Jones, R. Jeffrey Trapp, and Howard B. Bluestein. 2001. *Tornadoes and Tornadic Storms.* American Meteorological Society, Boston, MA, 167–221. https://doi.org/10.1007/978-1-935704-06-5_5

James W Deardorff. 1970. A numerical study of three-dimensional turbulent channel flow at large Reynolds numbers. *Journal of Fluid Mechanics* 41, 2 (1970), 453–480.

David M Gaffin and Stephen S Parker. 2006. A climatology of synoptic conditions associated with significant tornadoes across the southern Appalachian region. *Weather and forecasting* 21, 5 (2006), 735–751.

Francis R Hama. 1962. Progressive deformation of a curved vortex filament by its own induction. *The Physics of Fluids* 5, 10 (1962), 1156–1162.

Jorge Alejandro Amador Herrera, Jonathan Klein, Daoming Liu, Wojtek Pałubicki, Sören Pirk, and Dominik L. Michels. 2024. Cyclogenesis: Simulating Hurricanes and Tornadoes. *ACM Transaction on Graphics* 43, 4, Article 71 (7 2024). https://doi.org/10.1145/3658149

Doug L. James. 2020. Phong deformation: a better C0 interpolant for embedded deformation. *ACM Trans. Graph.* 39, 4, Article 56 (aug 2020), 9 pages. https://doi.org/10.1145/3386569.3392371

Joseph B Klemp. 1987. Dynamics of tornadic thunderstorms. *Annual review of fluid mechanics* 19, 1 (1987), 369–402.

Joel Koplik and Herbert Levine. 1993. Vortex reconnection in superfluid helium. *Phys. Rev. Lett.* 71 (Aug 1993), 1375–1378. Issue 9. https://doi.org/10.1103/PhysRevLett.71.1375

Anthony Leonard. 1985. Computing three-dimensional incompressible flows with vortex elements. *Annual Review of Fluid Mechanics* 17, 1 (1985), 523–559.

DC Lewellen, Baiyun Gong, and WS Lewellen. 2008. Effects of finescale debris on near-surface tornado dynamics. *Journal of the Atmospheric Sciences* 65, 10 (2008), 3247–3262.

DC Lewellen, WS Lewellen, and J Xia. 2000. The influence of a local swirl ratio on tornado intensification near the surface. *Journal of the atmospheric sciences* 57, 4 (2000), 527–544.

WS Lewellen. 1977. THEORETICAL MODELS OF THE TORNADO VORTEX. In *Proceedings of the Symposium on Tornadoes, Assessment of Knowledge and Implications for Man, June 22-24, 1976, Texas Tech University.* Institute for Disaster Research, Texas Tech University, 107.

WS Lewellen, DC Lewellen, and RI Sykes. 1997. Large-eddy simulation of a tornado's interaction with the surface. *Journal of the atmospheric sciences* 54, 5 (1997), 581–605.

W. Steve Lewellen. 1993. Tornado Vortex Theory. *Geophysical Monograph Series* 79 (Jan. 1993), 19–39. https://doi.org/10.1029/GM079p0019

Shiguang Liu, Zhangye Wang, Zheng Gong, and Qunsheng Peng. 2007. Real time simulation of a tornado. *The Visual Computer* 23, 8 (2007), 559–567.

Shiguang Liu, Zhangye Wang, Zheng Gong, and Qunsheng Peng. 2008. Simulation of atmospheric binary mixtures based on two-fluid model. *Graphical Models* 70, 6 (2008), 117–124.

Marcel Padilla, Albert Chern, Felix Knöppel, Ulrich Pinkall, and Peter Schröder. 2019. On bubble rings and ink chandeliers. *ACM Transactions on Graphics (TOG)* 38, 4 (2019), 1–14.

Louis Rosenhead. 1930. The spread of vorticity in the wake behind a cylinder. *Proceedings of the Royal Society of London. Series A, Containing papers of a mathematical and physical character* 127, 806 (1930), 590–612.

Richard Rotunno. 2013. The fluid dynamics of tornadoes. *Annual review of fluid mechanics* 45 (2013), 59–84.

P. G. Saffman. 1993. *Vortex Dynamics.* Cambridge University Press. https://doi.org/10.1017/CBO9780511624063

Martin A Satrio, David J Bodine, Anthony E Reinhart, Takashi Maruyama, and Franklin T Lombardo. 2020. Understanding how complex terrain impacts tornado dynamics using a suite of high-resolution numerical simulations. *Journal of the Atmospheric Sciences* 77, 10 (2020), 3277–3300.

Douglas G Schneider. 2009. The impact of terrain on three cases of tornadogenesis in the Great Tennessee Valley. *Electron. J. Oper. Meteor* 10, 11 (2009).

Klaus W. Schwarz. 1985. Three-dimensional vortex dynamics in superfluid He 4: Line-line and line-boundary interactions. *Physical Review B* 31, 9 (1985), 5782.

Vladimir Shtern, Anatoly Borissov, and Fazle Hussain. 1997. Vortex sinks with axial flow: Solution and applications. *Physics of Fluids* 9, 10 (1997), 2941–2959.

Michael K Tippett, Chiara Lepore, and Joel E Cohen. 2016. More tornadoes in the most extreme US tornado outbreaks. *Science* 354, 6318 (2016), 1419–1423.

A Yu Varaksin. 2017. Air tornado-like vortices: Mathematical modeling (a review). *High Temperature* 55 (2017), 286–309.

Aleksey Yu Varaksin and Sergei V Ryzhkov. 2023. Mathematical Modeling of Structure and Dynamics of Concentrated Tornado-like Vortices: A Review. *Mathematics* 11, 15 (2023), 3293.

Neil B. Ward. 1972. The Exploration of Certain Features of Tornado Dynamics Using a Laboratory Model. *Journal of Atmospheric Sciences* 29, 6 (1972), 1194 – 1204.

Steffen Weißmann and Ulrich Pinkall. 2010. Filament-based smoke with vortex shedding and variational reconnection. In *ACM SIGGRAPH 2010 papers.* 1–12.

A The necessary condition for Eq. (11) to satisfy the Navier-Stokes equation

While the Stokes stream function provides axial and radial velocities that automatically fulfill the continuity equation, the full solution to Eq. (11) may violate the Navier-Stokes equations. In cylindrical coordinates, the Navier-Stokes equation for an axisymmetric and steady flow of a constant viscosity reduces to [Acheson 1991, Appendix A.6]:

$$
\begin{cases}
v_r \dfrac{\partial v_r}{\partial r} + v_z \dfrac{\partial v_r}{\partial z} - \dfrac{v_\theta^2}{r} = \\
\qquad -\dfrac{1}{\rho}\dfrac{\partial p}{\partial r} + \mu\left(\dfrac{1}{r}\dfrac{\partial}{\partial r}\left(r\dfrac{\partial v_r}{\partial r}\right) + \dfrac{\partial^2 v_r}{\partial z^2} - \dfrac{v_r}{r^2}\right), & \text{(14a)} \\
v_r \dfrac{\partial v_\theta}{\partial r} + v_z \dfrac{\partial v_\theta}{\partial z} + \dfrac{v_\theta v_r}{r} = \mu\left(\dfrac{1}{r}\dfrac{\partial}{\partial r}\left(r\dfrac{\partial u_\theta}{\partial r}\right) - \dfrac{v_\theta}{r^2}\right), & \text{(14b)} \\
v_r \dfrac{\partial v_z}{\partial r} + v_z \dfrac{\partial v_z}{\partial z} = -\dfrac{1}{\rho}\dfrac{\partial p}{\partial z} + \mu\left(\dfrac{1}{r}\dfrac{\partial}{\partial r}\left(r\dfrac{\partial u_z}{\partial r}\right)\right), & \text{(14c)}
\end{cases}
$$

where ρ is the density of the flow, μ is the kinetic viscosity and p is the pressure. To begin with, it is easy to establish that Eq. (11) satisfies the azimuthal momentum in Eq. (14b). Thereafter, by differentiating Eq. (14a) in z and Eq. (14c) in r and subtracting the two, we arrive at the necessary condition for our vortex model to satisfy the Navier-Stokes equation:

$$
\begin{aligned}
\left(\mu\nabla_z^4 f\right) r^3 - \left(\alpha\nabla_z^3 f\right) r^2 + \left(2\alpha\nabla_z f\nabla_z^2 f\right) r \\
+ 3\alpha\nabla_z f + 3\mu = 0, \qquad \forall r > 0.
\end{aligned}
\tag{15}
$$

To ensure that the above equality holds, higher-order derivatives of f should vanish with only $\nabla_z f = -\mu/\alpha$ remaining. This means that f has to be a linear function if we restrict v_z to be proportional to $1/r$, which could be an unnecessarily strong requirement. In fact, this particular case matches Shtern's solution in Eq. (6) with constants $W_c = W_p = 0$ and Re $= -1$. Apparently, this is too restrictive in practice, and so we relax strict enforcement of the Navier-Stokes requirements in the interests of increased flexibility of user control.