

# Frequency-Domain Smoke Guiding

ZAHRA FOROOTANINIA, University of Minnesota

RAHUL NARAIN, Indian Institute of Technology Delhi

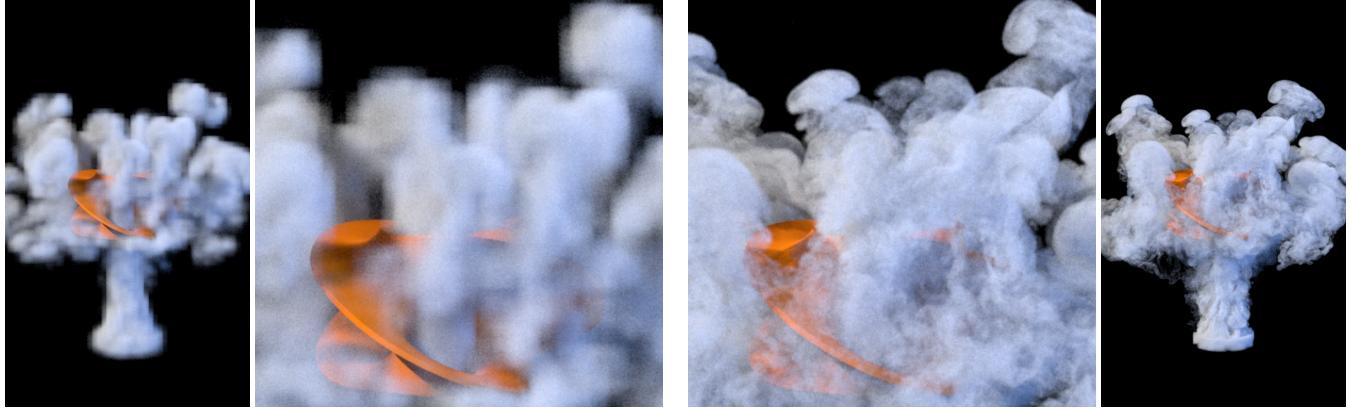


Fig. 1. Rising smoke flows around a complex obstacle, creating an interesting pattern of secondary plumes in this  $64 \times 64 \times 32$  simulation (left). Our method (right) constrains a  $512 \times 512 \times 256$  simulation to match the large-scale flow of the low-resolution input, so that we retain both the overall behavior of the original smoke and the turbulent fine-scale dynamics that arise from high-resolution fluid simulation.

We propose a simple and efficient method for guiding an Eulerian smoke simulation to match the behavior of a specified velocity field, such as a low-resolution animation of the same scene, while preserving the rich, turbulent details arising in the simulated fluid. Our method works by simply combining the high-frequency component of the simulated fluid velocity with the low-frequency component of the input guiding field. We show how to eliminate the grid-aligned artifacts that appear in naive guiding approaches, and provide a frequency-domain analysis that motivates the use of ideal low-pass and high-pass filters to prevent artificial dissipation of small-scale details. We demonstrate our method on many scenes including those with static and moving obstacles, and show that it produces high-quality results with very little computational overhead.

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

Additional Key Words and Phrases: fluid control, fluid guiding, fluid animation

#### ACM Reference Format:

Zahra Forootaninia and Rahul Narain. 2020. Frequency-Domain Smoke Guiding. *ACM Trans. Graph.* 39, 6, Article 172 (December 2020), 10 pages. <https://doi.org/10.1145/3414685.3417842>

Authors' addresses: Zahra Forootaninia, foro0012@umn.edu, University of Minnesota; Rahul Narain, narain@cse.iitd.ac.in, Indian Institute of Technology Delhi.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

© 2020 Association for Computing Machinery.

0730-0301/2020/12-ART172 \$15.00

<https://doi.org/10.1145/3414685.3417842>

## 1 INTRODUCTION

Realistic animation of smoke is a critical part of many visual effects applications, but still presents computational and artistic challenges due to the chaotic nature of fluid dynamics. High-resolution smoke simulation is expensive and requires a significant amount of computation time and memory. As such, artists prefer to work with low-resolution simulations when first designing a scene. Once the desired large-scale behavior is obtained, one would like to then scale up the resolution of the simulation for the final high-quality result, retaining the overall flow while better resolving small-scale turbulent details. However, as shown in Figure 2, increasing the resolution of the simulation affects the amount of artificial dissipation and other numerical effects, resulting in noticeably different behavior even at the large scales. Therefore, additional trials at high resolution may be needed to preserve the desired artistic intent.

To overcome this problem, we desire a method to *guide* a smoke simulation to follow a large-scale flow specified by the user, such as the result of a previously computed lower-resolution simulation. Many such techniques have been proposed in previous work [Huang and Keyser 2013; Huang et al. 2011; Inglis et al. 2017; Nielsen and Christensen 2010; Nielsen et al. 2009], which apply guiding forces or constraints on the simulated velocity field to stay as close as possible to the guide velocity field. However, all of these techniques add a significant overhead to the already high computational cost of unguided high-resolution simulation.

We present a simple technique for fluid guiding based on projection in the frequency domain. The key idea of our method is that the low-frequency component of a velocity field captures its overall large-scale behavior, so guiding is most naturally expressed as a hard constraint on the low frequencies alone. Therefore, our method proceeds as follows. At each time step, we extract the low-frequency

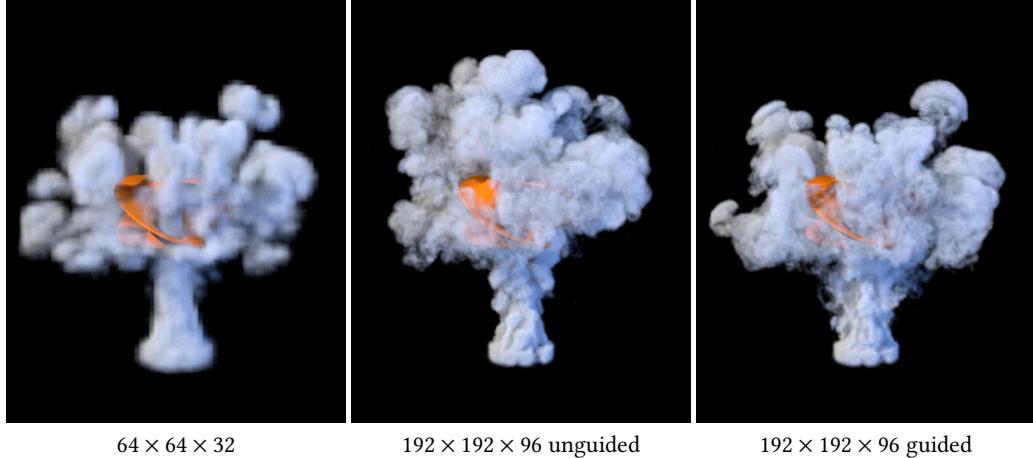


Fig. 2. The same scene as Fig. 1 but with 3× upscaling. Without guiding, a higher-resolution simulation exhibits substantially different behavior, even with a smaller upscaling factor.

component of the simulation using an ideal low-pass filter in the frequency domain, and constrain it to match the corresponding frequencies of the guiding velocity field. The high-frequency component of the simulated velocity field is unaffected by this process. In this way, the large-scale behavior of the result exactly matches that of the guiding field, while the detailed features of flow arising from high-resolution simulation are perfectly preserved. The amount of guiding can be controlled by changing the filtering cutoff.

Our method is straightforward and easy to implement on top of an existing grid-based fluid solver. It adds very little computational overhead, requiring only an FFT and IFFT at each time step. We show results on various smoke scenes including those with static and moving obstacles. In all cases, our method successfully preserves the large-scale motion of the guiding flow while adding rich, dynamic flow details in the small scales. Visual comparisons with unguided high-resolution simulations indicate that our method preserves the qualitative appearance of small-scale flow features to a better extent than the method of Inglis et al. [2017].

## 2 RELATED WORK

Fluid simulation is a well developed field in computer animation, and we refer readers to Bridson [2015] and Koschier et al. [2019] for a comprehensive discussion of grid-based and particle-based methods respectively. In this section, we review the most closely related work in controlling and guiding fluids to achieve desired artistic effects.

There are two main approaches for modifying a full-resolution fluid simulation to produce a desired behaviour, depending on how this behavior is specified. In *fluid control* methods, only a target shape or smoke distribution is given, and the fluid velocities must be controlled to drive the fluid towards the target state over time. In *fluid guiding* methods, the target velocity field itself is specified over the entire domain at all time instants, and the fluid is constrained to follow the given motion while retaining its physically plausible

small-scale flow details. Another alternative to fluid guiding is offered by *detail synthesis* techniques, which forgo high-resolution fluid simulation in favor of non-physical approaches such as procedural noise fields and generative neural networks.

### 2.1 Fluid control

Much work on fluid control has focused on driving simulated smoke or liquids to form specified shapes at chosen keyframes. Treuille et al. [2003] introduced a model to control smoke simulation by defining an objective function which measures how well the simulation matches the specified keyframes. McNamara et al. [2004] introduced an efficient method for this problem using gradient-based nonlinear optimization. Fattal and Lischinski [2004] instead add control forces proportional to the gradient of the target density distribution. Shi and Yu proposed methods of controlling the dynamics of smoke [Shi and Yu 2005a] and liquids [Shi and Yu 2005b] to follow rapidly changing targets. Barbic and Popović [2008] introduced a general method for controlling physics-based simulations using gentle forces, which can be applied to low-dimensional model-reduced fluids. Pan and Manocha [2017] provided a method for computing locally optimal control forces using an ADMM optimization.

To animate “sloppy” characters made of liquid, Rasmussen et al. [2004] use control particles defined on a user-specified surface to apply variable amounts of control on the fluid motion. Raveendran et al. [2012] similarly define a set of control meshes to track a desired shape, where the mesh velocities act as boundary conditions during the projection step and blended with the uncontrolled fluid velocities to control over the fluid.

Instead of forcing the fluid to create user-specified shapes, one may instead seek to make local changes to the result of a traditional fluid simulation while maintaining physically plausible behavior. Pan et al. [2013] present a method for locally editing liquid simulations via intuitive controls like sketching and dragging. Manteaux et al. [2016] allow liquid editing by automatically detecting space-time features like splashes so that the user can copy and paste them.

## 2.2 Fluid guiding

In fluid guiding, we wish to enhance an existing fluid motion by preserving its large-scale features, while at the same time generating new small-scale flow details. Guiding methods do so by constraining a high-resolution simulation to follow the specified fluid flow, while allowing secondary turbulent dynamics to arise naturally.

Thürey et al. [2006] proposed a detail-preserving fluid control method by creating artificial control particles that exert an attraction force on the fluid particles and push them toward the desired path. To ensure that the fluid velocity matches that of the control particles without disturbing the natural small-scale fluid motion, they separate the overall fluid motion from fine-scale detail using a low-pass filter on the velocity field.

Nielsen et al. [2009] formulated guiding as a constrained minimization problem in which the high-resolution velocity field is required to be as close as possible to an upsampled version of the input low-resolution field. In subsequent work, Nielsen and Christensen [2010] separated the low frequencies from high frequencies, making the mathematical formulation more efficient for time-dependent guiding of smoke simulation. Huang and colleagues proposed sampling-based techniques for fluid guiding [Huang and Keyser 2013; Huang et al. 2011], in which they match sampled data from the guiding flow at various positions and scales. This gives users the ability to identify key portions of the simulation to maintain. Inglis et al. [2017] proposed an optimization method for fluid simulation, in which fluid guiding is expressed via a soft constraint on the simulated velocity and the minimization is carried out via a primal-dual algorithm.

For guiding of liquid flow, Nielsen and Bridson [2011] construct a guide shape from a coarse simulation so that the high-resolution simulation can be run on a much shallower volume. Stomakhin and Selle [2017] introduced a similar method for particle-based simulations such as FLIP and SPH.

## 2.3 Detail Synthesis for Fluids

Many techniques have been proposed to add plausible fine-scale features directly on top of an input low-resolution fluid without additional numerical simulation. We refer to these as detail synthesis techniques.

One popular class of techniques is that of turbulence models, such as the popular wavelet turbulence method [Kim et al. 2008], which augment the coarse flow with synthetic details through procedural noise or texture synthesis. We refer the reader to Thurey et al. [2013] for a detailed survey, and note the more recent method of Sato et al. [2018] that transfers the style of one turbulent fluid to another. Recently, there has also been a growing trend of CNN-based techniques for super-resolving smoke simulations by synthesizing a high-resolution density field [Werhahn et al. 2019; Xie et al. 2018] or velocity field [Bai et al. 2019].

The advantage of detail synthesis techniques is that they avoid the computational cost of performing high-resolution fluid simulation. However, with the exception of the recent work of Bai et al. [2019], they are limited in their ability to reproduce the rich secondary dynamics of turbulent flows, since the added details do not necessarily obey the Navier-Stokes equations. In this work, we focus on fluid

guiding, and seek to remain as close to the physics of the fluid as possible.

## 3 SMOKE GUIDING

In this section, we will define the fluid guiding problem in more detail, and introduce a generic framework through which techniques for this problem may be viewed. We assume that an existing fluid simulator is available, which can advance the velocity field  $\mathbf{u}$  over one time step. In the fluid guiding problem, we are further given a user-specified, time-varying guide velocity field  $\mathbf{u}_g$ . Our goal is to modify the velocity field produced by the simulator, so that it follows the overall motion of  $\mathbf{u}_g$  in the large scales. Simultaneously, the result should not change the small scales of the simulated velocity, so that we retain all the turbulent details of fluid motion produced by the simulation. In summary, at every time step, we take the new velocity field computed by the fluid simulator, which we denote  $\mathbf{u}_s$ , and modify it according to the guiding field  $\mathbf{u}_g$  to get the guided velocity  $\mathbf{u}$ . The guided velocity is then used as the input for the fluid simulator at the next time step.

At this stage, the aim of fluid guiding is still defined only informally. We may formalize the problem by defining the large-scale motion of the fluid as the result of a low-pass filter  $L(\mathbf{u})$  applied to the velocity field, and the small-scale details as the remaining high-frequency component  $H(\mathbf{u}) = \mathbf{u} - L(\mathbf{u})$ . For example, Thürey et al. [2006] define  $L$  via a blur kernel centered on control particles in an SPH simulation; while most previous work on guiding grid-based fluids [Inglis et al. 2017; Nielsen and Christensen 2010; Nielsen et al. 2009] uses convolution with a Gaussian kernel,  $L(\mathbf{u}) = G * \mathbf{u}$ .

At this point, it is tempting to define the task of fluid guiding as finding a velocity field whose low-frequency component matches that of the guiding flow,

$$L(\mathbf{u}) = L(\mathbf{u}_g). \quad (1)$$

Such a constraint was considered by Nielsen et al. [2009], who pointed out that it would be numerically very badly behaved for most low-pass filters. In fact, for a Gaussian filter, it can be shown by a frequency-domain argument that in the continuous case, (1) has a unique solution  $\mathbf{u} = \mathbf{u}_g$ , leaving no room for  $\mathbf{u}$  to retain small-scale details from the simulation.<sup>1</sup> Nielsen et al. [2009] and Inglis et al. [2017] have also considered a simple guiding method

$$\mathbf{u} = H(\mathbf{u}_s) + L(\mathbf{u}_g), \quad (2)$$

where one simply combines the low-frequency component of the guiding flow with the high-frequency details from the simulation. However, both sets of authors have shown that this “detail-preserving blend”, inspired by the particle-based guiding method of Thürey et al. [2006], produces severe artifacts in a grid-based setting. Therefore, a series of works [Inglis et al. 2017; Nielsen and Christensen 2010; Nielsen et al. 2009] have proposed more sophisticated techniques where the guiding operation is expressed as a large-scale optimization problem.

In the following section, we analyze detail-preserving blending in the frequency domain and show that existing guiding methods based

<sup>1</sup>In the frequency domain, convolution  $G * \mathbf{u}$  becomes a pointwise multiplication  $\hat{G}\hat{\mathbf{u}}$  at each frequency, so (1) becomes  $\hat{G}\hat{\mathbf{u}} = \hat{G}\hat{\mathbf{u}}_g$ . Since the Fourier transform  $\hat{G}$  is also a Gaussian and therefore nonzero everywhere, we can divide it out to get  $\hat{\mathbf{u}} = \hat{\mathbf{u}}_g$ .

on Gaussian filters result in artificial dissipation of flow features. Then we show that by simply changing the low-pass filter  $L$ , one can avoid this drawback, and in fact obtain a method that exactly satisfies (1).

### 3.1 Frequency-domain analysis of fluid guiding

The low-frequency and high-frequency components of the velocity field as described above can be analyzed more easily in the frequency domain, since the convolution becomes a simple pointwise multiplication. We adopt the convention that Fourier transform of any quantity  $x$  is denoted  $\hat{x}$ .

For now, let us consider the choice  $L(\mathbf{u}) = G * \mathbf{u}$ . Then the low-frequency component of the velocity field is obtained in the frequency domain as  $\hat{L}\hat{\mathbf{u}} = \hat{G}\hat{\mathbf{u}}$ . Similarly, the high-frequency component is  $\hat{H}\hat{\mathbf{u}} = (1 - \hat{G})\hat{\mathbf{u}}$ .

The guiding field  $\mathbf{u}_g$  is typically very smooth, since it arises from a smooth user input or from a low-resolution simulation with significant numerical dissipation. After detail-preserving blending, we obtain

$$\hat{\mathbf{u}} = \hat{G}\hat{\mathbf{u}}_g + (1 - \hat{G})\hat{\mathbf{u}}_s.$$

The above equation shows that all frequency modes of the simulated velocity field are reduced by a factor of  $1 - \hat{G}$  every time step. Since  $\hat{G}$  is itself a Gaussian, and is always positive, this indicates that detail-preserving blending introduces artificial dissipation, gradually damping out the flow features of the simulated fluid (although unlike viscosity, the rate of damping is lower for higher frequencies). In fact, such a guiding method is also time step dependent, since if the time step is reduced, more guiding operations will be applied and the flow will dissipate faster. A similar frequency-domain analysis can be carried out for previous methods [Inglis et al. 2017; Nielsen and Christensen 2010; Nielsen et al. 2009] if the guiding parameters are spatially uniform. We show this analysis in Appendix A.

The dissipation that results from the Gaussian filter is due to the fact that it is not *idempotent*:  $\hat{G}^2 \neq \hat{G}$ , and consequently  $(1 - \hat{G})^2 \neq (1 - \hat{G})$ . As a result, multiple applications of  $1 - \hat{G}$  keep attenuating the features of the fluid velocity. To avoid this dissipation and preserve all the desired details of the simulated velocity field, it is necessary for the guiding process to be idempotent: that is, applying guiding multiple times on the simulated flow should not change the result beyond the initial guiding. This is easy to achieve in the frequency domain: the coefficients of the filters  $\hat{L}$  and  $\hat{H}$  must be 0 or 1. Therefore, we choose them to be *ideal* low-pass and high-pass filters respectively,

$$\hat{L}(v) = \begin{cases} 1 & \text{if } \|v\| < v_{\text{cutoff}} \\ 0 & \text{otherwise,} \end{cases} \quad (3)$$

$$\hat{H}(v) = \begin{cases} 0 & \text{if } \|v\| < v_{\text{cutoff}} \\ 1 & \text{otherwise.} \end{cases} \quad (4)$$

Here the parameter  $v_{\text{cutoff}}$  is the cut-off frequency of the filters. As a result,  $\hat{\mathbf{u}} = \hat{L}\hat{\mathbf{u}}_g + \hat{H}\hat{\mathbf{u}}_s$  is the velocity field obtained by replacing the all low-frequency (i.e. below  $v_{\text{cutoff}}$ ) coefficients of  $\mathbf{u}_s$  with the corresponding coefficients of  $\mathbf{u}_g$ . The frequency cutoff affects the amount of guiding. As seen in Figures 8 and 9, the higher the cutoff,

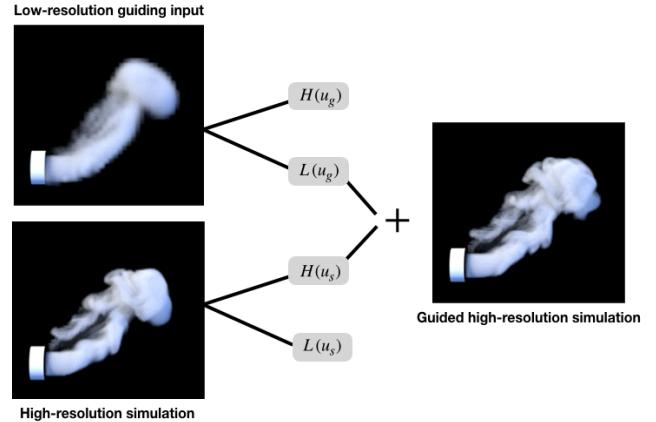


Fig. 3. Our method takes a guide velocity field as an input and the current velocity from the fluid simulation, and superimposes the low frequencies of the guide velocity with the high frequencies of the current velocity. This yields the new guided velocity field, which will be used as the current velocity for the next frame of the simulation.

the more guiding we obtain, and the more the overall flow will resemble the guiding velocity field.

We can also verify that the guiding operation is idempotent. If we denote  $k$  iterations of guiding via  $\hat{\mathbf{u}}^k = \hat{L}\hat{\mathbf{u}}_g + \hat{H}\hat{\mathbf{u}}^{k-1}$ , then

$$\begin{aligned} \hat{\mathbf{u}}^k &= \hat{L}\hat{\mathbf{u}}_g + \hat{H}\hat{\mathbf{u}}^{k-1} \\ &= \hat{L}\hat{\mathbf{u}}_g + \hat{H}(\hat{L}\hat{\mathbf{u}}_g + \hat{H}\hat{\mathbf{u}}^{k-2}) \\ &= \hat{L}\hat{\mathbf{u}}_g + \hat{H}\hat{\mathbf{u}}^{k-2} \\ &\vdots \\ &= \hat{L}\hat{\mathbf{u}}_g + \hat{H}\hat{\mathbf{u}}^0 \end{aligned} \quad (5)$$

because  $\hat{L}\hat{H} = 0$  and  $\hat{H}^2 = \hat{H}$ . As a result no matter how many guiding operations are applied, we always can preserve the same amount of details that we could preserve after the first time. Finally, the result of guiding exactly satisfies the guiding constraint (1), since  $\hat{L}\hat{\mathbf{u}} = \hat{L}(\hat{L}\hat{\mathbf{u}}_g + \hat{H}\hat{\mathbf{u}}_s) = \hat{L}\hat{\mathbf{u}}_g$ .

### 3.2 Implementation of fluid guiding with ideal filters

We apply our guiding procedure to the simulated velocity by taking the Fourier transform of both guiding and simulated velocity fields. Since these velocity fields are not periodic but rather are subject to boundary conditions at the grid boundaries, one has to make a careful choice of how to compute the Fourier modes. Following Long and Reinhard [2009], we assume no-through, free-slip wall boundary conditions, and use the discrete cosine/sine transforms (DCT/DST) to obtain Fourier modes that respect the boundaries. These transforms are applied independently to each component of the velocity field and independently in each direction, with the appropriate type of DCT/DST chosen to account for the staggered nature of the MAC grid and the even/odd symmetry imposed by the boundary conditions.

**Algorithm 1** Fluid simulation with frequency-domain guiding

---

```

Run fluid simulator on  $\mathbf{u}$  without final pressure projection
to get  $\mathbf{u}_s$ 
Perform DCT/DST to get  $\hat{\mathbf{u}}_s$  and  $\hat{\mathbf{u}}_g$ 
Set  $\hat{\mathbf{u}} = \hat{L}\hat{\mathbf{u}}_g + \hat{H}\hat{\mathbf{u}}_s$ 
Perform IDCT/IDST to get  $\mathbf{u}$ 
Perform pressure projection on  $\mathbf{u}$ 

```

---

For completeness, we explain these choices here. Consider the  $x$ -component of the velocity field,  $u_x$ . No-through boundary conditions require  $u_x = 0$  at the left and right walls of the domain, so  $u_x$  should be treated as an odd function of  $x$ , and we use the DST-I variant of the discrete sine transform along the  $x$  axis of the grid. With free-slip boundaries,  $u_x$  is most naturally viewed as an even function of  $y$  and  $z$ , and since the samples are offset by half a grid cell in the MAC grid, a DCT-II transform is appropriate. Similarly for  $u_y$  and  $u_z$ , we use DST-I in the direction aligned with the component and DCT-II in the other two.

We have used this scheme for all our examples, although most of them feature open boundary conditions instead; nevertheless, we have not observed any significant artifacts from this choice. Cui et al. [2018] have described compatible Fourier transforms for domains having some or all boundaries open; we wish to adopt these in future work.

Having transformed both velocity fields from guide and simulation into the frequency domain, we can apply the ideal low- and high-pass filters to  $\mathbf{u}_g$  and  $\mathbf{u}_s$ . In fact, forming the guided velocity field simply amounts to copying the values of all Fourier coefficients lower than the cutoff from the guiding velocity field to the simulated velocity field. After applying the guiding, we take the inverse DCT/DST of the final coefficients with appropriate boundary conditions to get the new guided velocities back.

A convenient property of the frequency domain is that for a rectangular domain, the divergence-free constraint on the velocity field acts independently on each frequency mode [Long and Reinhard 2009; Stam 2002]. Since our guiding scheme simply transfers coefficients from selected modes in  $\mathbf{u}_g$  independently to corresponding modes in  $\mathbf{u}_s$ , it is naturally compatible with the divergence-free constraint, in the sense that if both  $\mathbf{u}_g$  and  $\mathbf{u}_s$  are divergence-free, then the result of guiding  $\mathbf{u}$  is automatically divergence-free as well. In fact, guiding and pressure projection are interchangeable: projecting an arbitrary velocity field to be divergence-free and then performing guiding gives the same result as guiding the original velocity field and then projecting the result.

This is no longer true if the fluid domain is not rectangular. In particular, in the presence of obstacles, the fluid velocity field is no longer defined everywhere on the simulation grid. We extend the domain of the velocity field by filling all obstacle cells with the obstacle velocity, and perform guiding on the full grid without modification as if there were no obstacles. Alternatively, one could extrapolate the fluid velocity into obstacles, although we have not tried this yet. In either case, however, since the guiding operation is not aware of the obstacles, the guided fluid may no longer respect the obstacle boundaries and thus incompressibility may not be guaranteed. In

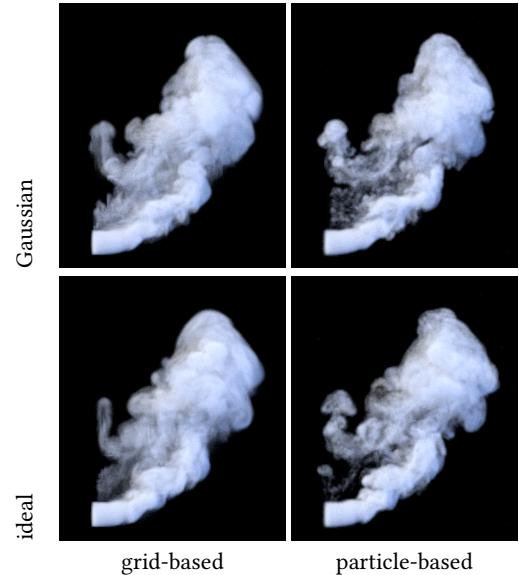


Fig. 4. Both Gaussian blending (top) and our ideal frequency-domain guiding (bottom) exacerbate axis-aligned discretization artifacts in grid-based density advection (left). Tracking densities by advecting tracer particles (right) eliminates these artifacts.

this case, we would require an additional pressure solve to recover a divergence-free velocity field on the fluid domain. This additional cost can be avoided by observing that pressure projection is invariably the last step in the fluid simulator, and we know that guiding and pressure projection are interchangeable in the obstacle-free case. So we simply move the pressure projection after the guiding step: that is, we run one timestep of the fluid simulator *without* the final pressure projection, perform guiding, and then apply pressure projection afterwards. This yields a valid divergence-free velocity field for scenes with obstacles, while causing no change to the results in the absence of obstacles. Furthermore, it can be seen that our obstacle handling approach does not incur any additional computational overhead.

The steps of our algorithm are summarized in Algorithm 1. Figure 3 illustrates the steps above as well.

### 3.3 Grid Artifacts

Our initial experiments with fluid guiding encountered an unexpected problem: purely grid-based smoke simulation with guiding often results in streak-like artifacts aligned with the grid axes (Fig. 4). These are especially noticeable in Gaussian detail-preserving blending, in which context they were referred to as “frosted glass” artifacts by Inglis et al. [2017]; unfortunately, they can also be observed in our ideal guiding scheme. After much investigation, we found that these were discretization artifacts caused by higher-order grid-based advection schemes. Switching from MacCormack advection [Selle et al. 2008] to first-order semi-Lagrangian advection reduces the artifacts, and further reverting the spatial interpolation to linear instead of monotone cubic eliminates them. We hypothesize that the reason these artifacts are exacerbated by guiding is that the guiding

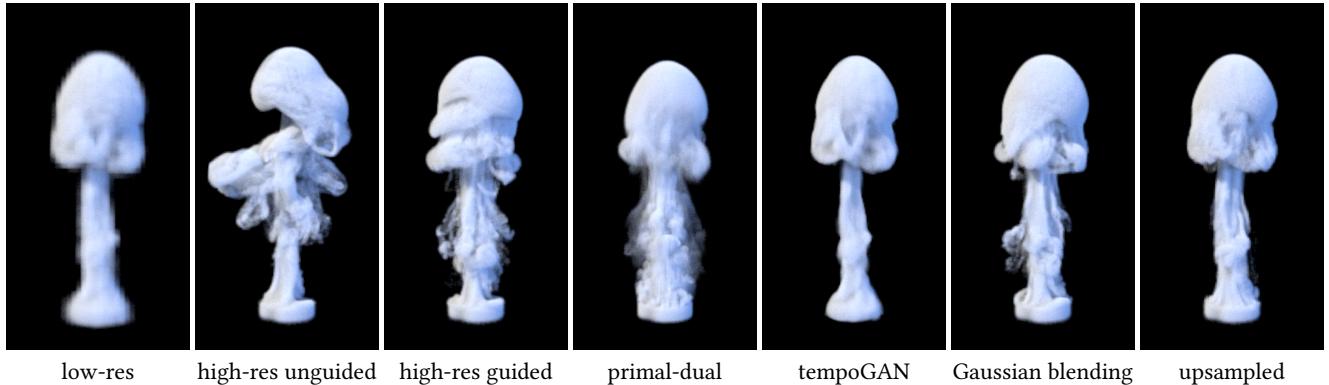


Fig. 5. A plume of smoke is emitted from a cylindrical source, becoming turbulent as it rises. From left, we show results from a low-resolution simulation, a high-resolution unguided simulation, our guiding method, the primal-dual guiding method [Inglis et al. 2017], tempoGAN [Xie et al. 2018], Gaussian detail-preserving blending, and high-res density advection using upsampled velocities. All higher-resolution results are 3×, except tempoGAN which performs 4× super-resolution.

process forces many regions of the domain to near-zero velocities, which is a difficult case for higher-order semi-Lagrangian advection. Unfortunately, with low-order advection on grids, the result is too diffusive to be useful.

Instead, we track density using particles: at each time step, we seed particles at smoke sources and advect them passively using the current velocity field, then accumulate them onto a grid to obtain the smoke density that is used for buoyancy forces and rendering. In the seeding step, we add new particles to each grid cell proportional to the change in density due to the smoke source, up to a maximum of  $3^3$  particles per grid cell. Particles are advected with RK4 time integration and trilinear spatial interpolation on the velocity grid. The smoke density on the grid is computed by accumulating particle counts with linear interpolation weights. We do not make any effort to avoid particle clumping.

We use this scheme for all of our results other than the left column of Fig. 4. For visualization, we found it expedient to render the grid density rather than the particles themselves, although this gives some of the low-resolution results a voxelized appearance. All other aspects of the fluid simulator are unchanged, including the velocity self-advection step which still uses MacCormack advection on a MAC grid like before. One could certainly go further and switch to a particle-in-cell scheme like FLIP [Brackbill and Ruppel 1986; Zhu and Bridson 2005] or APIC [Jiang et al. 2015] for velocity advection as well, but we chose to remain as close as possible to a purely grid-based smoke solver.

## 4 RESULTS

We have implemented our method in the Mantaflow fluid simulation code [Thurey and Pfaff 2016]. We use the second-order advection-reflection method [Narain et al. 2019; Zehnder et al. 2018] for all examples except those in Figure 5 which have been used for comparisons. In the latter, we compare our method with Inglis et al. [2017], so we use the traditional advection-projection method to be consistent with their results. For all our guided simulations, we choose  $v_{\text{cutoff}} = 1/4$  unless stated otherwise. These cutoff values are

normalized to the grid size, that is,  $v_{\text{cutoff}} = 1/k$  means that guiding is imposed on modes with a half-period of  $k$  low-resolution grid cells or more.

We measured the performance of our method on a PC with a six-core 3.50GHz Intel Xeon CPU. The dimensions of the low-resolution input and the high-resolution simulation along with computation time for all our results are given in Table 1.

*Rising plume.* As shown in Figure 5, we simulated a plume of smoke from a cylindrical source rising due to buoyancy. Even in this simple scene, the low-resolution and high-resolution simulations diverge significantly due to the turbulent vortices emerging as the plume rises. Our method, applied using the low-resolution result as the guiding field, matches the overall shape of the plume from the low-resolution input, but preserves the qualitative appearance of turbulent vortices seen in the unguided high-resolution simulation.

We also ran the primal-dual guiding method of Inglis et al. [2017] with  $w = 1$  and  $\beta = 4$  on the same example. We found that it matches the overall shape of the guiding input very well, but the appearance of the resulting flow differs qualitatively from the reference high-resolution simulation: it exhibits stronger small-scale details, such as near the base of the plume, and weaker mid-scale features.

To compare with the tempoGAN super-resolution model [Xie et al. 2018], we applied their method with our low-resolution smoke densities and velocities as input. In the resulting animation, the smoke distribution from the original simulation has been made sharper, but it is missing the small-scale features and vortices one expects to see from a high-resolution simulation.

We also compared our results with Gaussian detail-preserving blending as discussed in Section 3.1. Although with this approach the coarse shape of the plume is well preserved, the fine-scale flow details are weaker than with ideal guiding and primal-dual guiding due to the additional dissipation caused by blending. Finally, we show the result of simply upscaling the guiding velocity and performing high-resolution density advection. As expected, the smoke follows the guiding velocity with a sharp appearance, but no details get generated due to the lack of high-frequency dynamics.

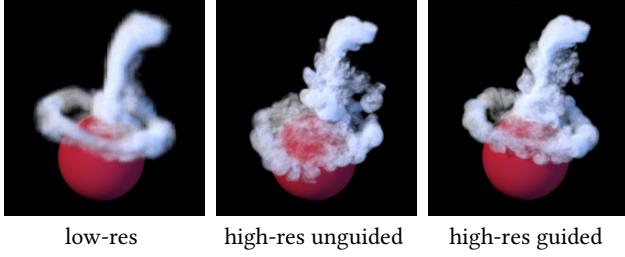


Fig. 6. Heavy smoke is poured on top of a sphere.

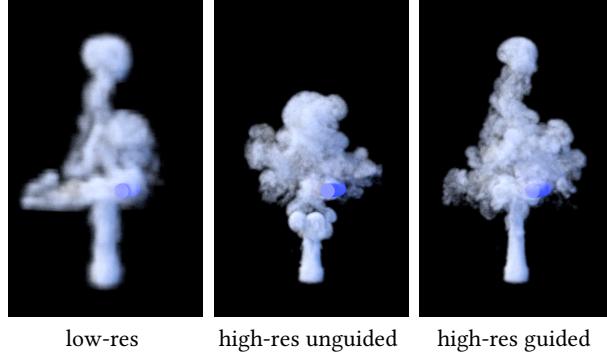


Fig. 7. A rising plume is interrupted by a moving cylinder.

*Static obstacles.* In Figure 6, we pour heavy smoke on top of a solid obstacle. The difference in resolutions changes the shape that the smoke forms around the sphere; the low-resolution simulation produces a noticeable vortex ring whereas in the high-resolution simulation the ring quickly breaks apart due to turbulence. By applying our guiding method, we can maintain the shape of the vortex ring while preserving the detailed flow appearance of the high-resolution simulation.

Guiding can make a more significant difference when the smoke interacts with complex objects having detailed geometry, since differences in obstacle discretization and boundary interactions can change the dynamics of the fluid dramatically. To illustrate this, we run our guiding method with an obstacle with complex topology and multiple holes, shown with 3 $\times$  increased resolution in Figure 2 and 8 $\times$  increased resolution in Figure 1. At low resolution, much of the smoke is blocked by the obstacle while coherent secondary plumes emerge from its holes; at high resolution the smoke moves easily around it resulting in a turbulent flow. Our guided result preserves the coherent behavior of the low-resolution simulation.

*Moving obstacle.* To demonstrate our method in the presence of a moving obstacle, we simulated a scenario in which a cylindrical obstacle is moving left to right sinusoidally above a source of smoke (Figure 7). As the smoke rises, its vertical motion is disrupted when the obstacle passes through it. The guided simulation retains the small plume escaping vertically above the cylinder. Simultaneously, it exhibits turbulent flow qualitatively similar to that of the high-res simulation in the neighbourhood of the obstacle.

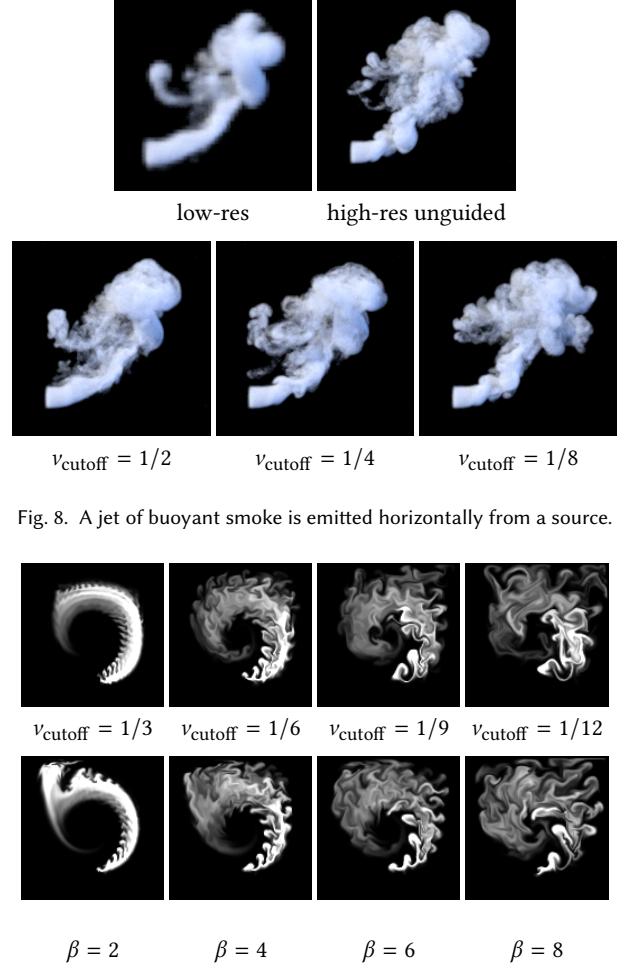
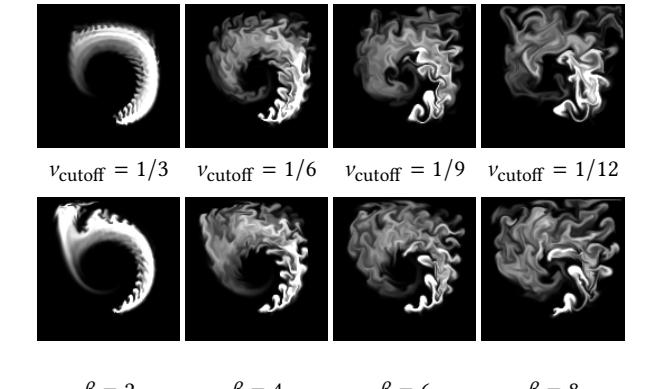


Fig. 8. A jet of buoyant smoke is emitted horizontally from a source.

Fig. 9. Comparison of our guiding method (top) to primal-dual guiding [Inglis et al. 2017] with  $W = 1$  (bottom). We get visually similar results when choosing parameters  $v_{\text{cutoff}} = \frac{2}{3\beta}$ .

*Buoyant jet.* We use this scenario to show the effect of different choices of the filters' frequency cutoff  $v_{\text{cutoff}}$ . We ran a simulation of an initially horizontal jet of smoke rising under buoyancy, shown in Figure 8. Guided simulations with different choices of cutoff are shown. The strength of the guiding lessens as the cutoff frequency gets smaller, since more frequency modes are unconstrained.

*Comparison with primal-dual fluid guiding.* For another comparison with the method of Inglis et al. [2017], we replicated their example of a 2D smoke simulation guided by a constant circular velocity field. The results are shown in Figure 9 for both our method with different values of  $v_{\text{cutoff}}$  and the primal-dual method with different values of their parameter  $\beta$ . We have experimentally observed that choosing  $v_{\text{cutoff}} = 2/(3\beta)$  produces fairly similar results between the two methods. Our method achieves qualitatively similar results to the primal-dual method, while our computational cost is an order of magnitude smaller, as shown in Table 2. However, the primal-dual method provides two parameters for guiding,  $W$  which

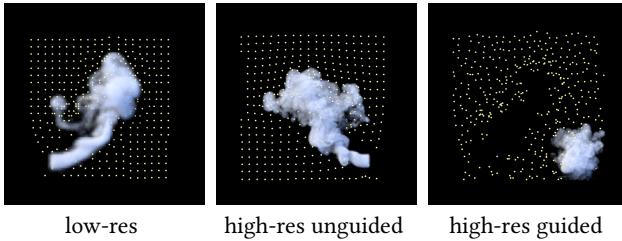


Fig. 10. Guiding using mismatched scenes for the low-res input and high-res simulation. Passive tracer particles are added to visualize the flow in smoke-free regions.

defines the guiding weight and  $\beta$  which controls the amount of smoothing. In comparison, our frequency-domain method has only one parameter, the frequency cutoff  $v_{\text{cutoff}}$ . Our method also only performs guiding globally across the entire fluid, while the primal-dual method permits both the guiding weight and the smoothing radius to vary across the domain. We discuss this limitation of our method further in the following section.

*Mismatch between input and simulation.* In all the above experiments, we have used the same scene setup for both the low-resolution input and the guided simulation, so that the flows are roughly similar. We still cannot guarantee perfect alignment due to discrepancies in the discretization of sources or obstacles, but we have not observed any objectionable artifacts due to this. However, to explore how guiding is affected by such differences, we show in Figure 10 an exaggerated example where the scenes are completely different, with smoke sources on opposite sides of the scene. Even in this case, the results appear largely plausible, with the smoke in the guided simulation churning in place as though blown by an external wind field. However, we also observe some extraneous motion of tracers in quiescent regions. These likely originate from “ringing” artifacts (related to the Gibbs phenomenon) caused by hard cutoffs in frequency-domain filtering.

## 5 DISCUSSION

*Performance.* The proposed frequency-domain method of smoke guiding is fast without sacrificing simulation quality. In Table 1 we show that the overhead of our method is very small compared to the cost of high-resolution simulation, since it only requires a Fourier transform and an inverse Fourier transform of the velocity field. This cost is comparable to that of the advection routines, and dominated by the cost of pressure projection. In particular, for the plume example (Figure 5), the breakdown of computational cost for the individual steps of the simulation was as follows: velocity advection: 16%, particle advection: 8%, pressure projection: 64%, guiding: 12%. Another aspect keeping our overhead low is that even in the presence of obstacles, we do not require an additional pressure solve.

*Constrained dynamics interpretation.* Guiding with ideal filters can be viewed as a projection operation, through which the current velocity field is projected onto the space of velocity fields whose low-frequency component equals that of the guide velocity  $\mathbf{u}_g$ . This

is similar in spirit to the TRACKS method [Bergou et al. 2007], which controls the dynamics of wrinkles and folds in thin shells by imposing constraints on the equations of motion based on the input guide shapes. In particular, as discussed in Section 3.1, we can express our guiding formulation as a hard constraint on the fluid motion,

$$\hat{\mathbf{L}}\hat{\mathbf{u}} = \hat{\mathbf{L}}\hat{\mathbf{u}}_g. \quad (6)$$

Consequently, the steps of our guiding algorithm can be interpreted as a projection method for time integration of a particular constrained dynamical system, namely a fluid subject to a conjunction of the Navier-Stokes or Euler equations and this additional guiding constraint.

*Limitations.* Our most significant limitation compared to previous work is that our guiding operation necessarily acts globally over the entire fluid domain. It is not possible at present to guide only a selected local region of the fluid while permitting the rest of the fluid to evolve in an unconstrained manner. For the same reason, the guiding velocity field must be specified everywhere in the domain, which is reasonable for the application of refining low-resolution preview simulations, but may be an onerous requirement for artist-designed guiding flow fields.

Closely related to the above limitation is the fact that our method is not suited for guiding liquid simulations. For liquids, the fluid domain is typically a small subset of the simulation grid and its boundary varies significantly over time. Thus a Fourier-based technique cannot be applied in this context.

Since our core guiding operation is unaware of obstacles, we apply it immediately before the pressure projection so that the final velocity field still remains incompressible and respects obstacle boundaries. However, in this case guiding and pressure projection do not commute and we may no longer satisfy the ideal guiding constraint (1). In principle, obstacles and irregular domains could be handled more naturally using Laplacian eigenfunctions [De Witt et al. 2012], which would give us Fourier-type modes that respect obstacle boundaries upon which guiding could be performed.

## REFERENCES

- Kai Bai, Wei Li, Mathieu Desbrun, and Xiaopei Liu. 2019. Dynamic Upsampling of Smoke through Dictionary-based Learning. arXiv:cs.GR/1910.09166
- Jernej Barbić and Jovan Popović. 2008. Real-time control of physically based simulations using gentle forces. *ACM transactions on graphics (TOG)* 27, 5 (2008), 1–10.
- Miklós Bergou, Saurabh Mathur, Max Wardetzky, and Eitan Grinspun. 2007. TRACKS: Toward Directable Thin Shells. In *ACM SIGGRAPH 2007 Papers (SIGGRAPH '07)*. ACM, New York, NY, USA, Article 50. <https://doi.org/10.1145/1275808.1276439>
- J.U. Brackbill and H.M. Ruppel. 1986. FLIP: A method for adaptively zoned, particle-in-cell calculations of fluid flows in two dimensions. *J. Comput. Phys.* 65, 2 (1986), 314 – 343. [https://doi.org/10.1016/0021-9991\(86\)90211-1](https://doi.org/10.1016/0021-9991(86)90211-1)
- Robert Bridson. 2015. *Fluid simulation for computer graphics*. AK Peters/CRC Press.
- Qiaodong Cui, Pradeep Sen, and Theodore Kim. 2018. Scalable Laplacian Eigenfluids. *ACM Trans. Graph.* 37, 4, Article 87 (July 2018), 12 pages. <https://doi.org/10.1145/3197517.3201352>
- Tyler De Witt, Christian Lessig, and Eugene Fiume. 2012. Fluid Simulation Using Laplacian Eigenfunctions. *ACM Trans. Graph.* 31, 1, Article 10 (Feb. 2012), 11 pages. <https://doi.org/10.1145/2077341.2077351>
- Raanan Fattal and Dani Lischinski. 2004. Target-driven Smoke Animation. *ACM Trans. Graph.* 23, 3 (Aug. 2004), 441–448. <https://doi.org/10.1145/1015706.1015743>
- R. Huang and J. Keyser. 2013. Automated sampling and control of gaseous simulations. 29 (2013), 751–760. <https://doi.org/10.1007/s00371-013-0798-0>
- Ruoguan Huang, Zeki Melek, and John Keyser. 2011. Preview-Based Sampling for Controlling Gaseous Simulations. In *Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA '11)*. Association

Table 1. Average simulation time per frame (in seconds) for all the 3D examples.

Scenario	Low Resolution	Sim Time	Sim Resolution	Unguided Sim Time	Guided Sim Time
Knot (Fig. 2)	$64 \times 64 \times 32$	0.97	$192 \times 192 \times 96$	21.89	24.5
Plume (Fig. 5)	$32 \times 64 \times 32$	0.41	$96 \times 192 \times 96$	5.15	5.87
Sphere (Fig. 6)	$64 \times 64 \times 32$	1.11	$192 \times 192 \times 96$	25.89	26.75
Moving cylinder (Fig. 7)	$32 \times 64 \times 32$	0.66	$96 \times 192 \times 96$	13.0	13.99
Buoyant jet (Fig. 8)	$64 \times 64 \times 32$	1.04	$192 \times 192 \times 96$	20.78	21.65

Table 2. Average simulation time per frame for a  $96 \times 192 \times 96$  guided simulation of the plume example (Figure 5) using our method and primal-dual guiding [Inglis et al. 2017].

Ideal guiding	$v_{\text{cutoff}} = 1/3$	$v_{\text{cutoff}} = 1/6$	$v_{\text{cutoff}} = 1/9$	$v_{\text{cutoff}} = 1/12$
	5.88	5.85	5.82	5.8
Primal-dual guiding	$\beta = 2$	$\beta = 4$	$\beta = 6$	$\beta = 8$
W=1	50.63	56.28	56.64	64.15
W=3	64.82	70.68	76.88	85.86

- for Computing Machinery, New York, NY, USA, 177–186. <https://doi.org/10.1145/2019406.2019430>
- Tiffany Inglis, M-L Eckert, James Gregson, and Nils Thuerey. 2017. Primal-Dual Optimization for Fluids. In *Computer Graphics Forum*, Vol. 36. Wiley Online Library, 354–368.
- Chenfanfu Jiang, Craig Schroeder, Andrew Selle, Joseph Teran, and Alexey Stomakhin. 2015. The Affine Particle-in-Cell Method. *ACM Trans. Graph.* 34, 4, Article 51 (July 2015), 10 pages. <https://doi.org/10.1145/2766996>
- Theodore Kim, Nils Thuerey, Doug James, and Markus Gross. 2008. Wavelet turbulence for fluid simulation. In *ACM Transactions on Graphics (TOG)*, Vol. 27. ACM, 50.
- Dan Koschier, Jan Bender, Barbara Solenthaler, and Matthias Teschner. 2019. Smoothed Particle Hydrodynamics Techniques for the Physics Based Simulation of Fluids and Solids. In *Eurographics 2019 - Tutorials*, Wenzel Jakob and Enrico Puppo (Eds.). The Eurographics Association. <https://doi.org/10.2312/egt.20191035>
- Benjamin Long and Erik Reinhard. 2009. Real-time fluid simulation using discrete sine/cosine transforms. In *Proceedings of the 2009 symposium on Interactive 3D graphics and games*. ACM, 99–106.
- Pierre-Luc Manteaux, Ulysse Vimont, Chris Wojtan, Damien Rohmer, and Marie-Paule Cani. 2016. Space-time Sculpting of Liquid Animation. In *Proceedings of the 9th International Conference on Motion in Games (MIG '16)*. ACM, New York, NY, USA, 61–71. <https://doi.org/10.1145/2994258.2994261>
- Antoine McNamara, Adrien Treuille, Zoran Popović, and Jos Stam. 2004. Fluid Control Using the Adjoint Method. *ACM Trans. Graph.* 23, 3 (Aug. 2004), 449–456. <https://doi.org/10.1145/1015706.1015744>
- Rahul Narain, Jonas Zehnder, and Bernhard Thomaszewski. 2019. A Second-Order Advection-Reflection Solver. *Proc. ACM Comput. Graph. Interact. Tech.* 2, 2, Article Article 16 (July 2019), 14 pages. <https://doi.org/10.1145/3340257>
- Michael B. Nielsen and Robert Bridson. 2011. Guide Shapes for High Resolution Naturalistic Liquid Simulation. *ACM Trans. Graph.* 30, 4, Article 83 (July 2011), 8 pages. <https://doi.org/10.1145/2010324.1964978>
- Michael B Nielsen and Brian B Christensen. 2010. Improved variational guiding of smoke animations. In *Computer Graphics Forum*, Vol. 29. Wiley Online Library, 705–712.
- Michael B Nielsen, Brian B Christensen, Nafees Bin Zafar, Doug Roble, and Ken Museth. 2009. Guiding of smoke animations through variational coupling of simulations at different resolutions. In *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. ACM, 217–226.
- Zherong Pan, Jin Huang, Yiyang Tong, Changxi Zheng, and Hujun Bao. 2013. Interactive Localized Liquid Motion Editing. *ACM Trans. Graph.* 32, 6, Article 184 (Nov. 2013), 10 pages. <https://doi.org/10.1145/2508363.2508429>
- Zherong Pan and Dinesh Manocha. 2017. Efficient Solver for Spacetime Control of Smoke. *ACM Trans. Graph.* 36, 4, Article 68a (July 2017). <https://doi.org/10.1145/3072959.3016963>
- Nick Rasmussen, Douglas Enright, Duc Nguyen, Sebastian Marino, Nigel Sumner, Willi Geiger, Samir Hoon, and Ronald Fedkiw. 2004. Directable photorealistic liquids. In *Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation*, 193–202.
- Karthik Raveendran, Nils Thuerey, Chris Wojtan, and Greg Turk. 2012. Controlling Liquids Using Meshes. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA '12)*. Eurographics Association, Goslar Germany, Germany, 255–264. <http://dl.acm.org/citation.cfm?id=2422356.2422393>
- Syuhei Sato, Yoshinori Dobashi, Theodore Kim, and Tomoyuki Nishita. 2018. Example-based Turbulence Style Transfer. *ACM Trans. Graph.* 37, 4, Article 84 (July 2018), 9 pages. <https://doi.org/10.1145/3197517.3201398>
- Andrew Selle, Ronald Fedkiw, Byungmoon Kim, Yingjie Liu, and Jarek Rossignac. 2008. An Unconditionally Stable MacCormack Method. *J. Sci. Comput.* 35, 2a–3 (June 2008), 350–371. <https://doi.org/10.1007/s10915-007-9166-4>
- Lin Shi and Yizhou Yu. 2005a. Controllable smoke animation with guiding objects. *ACM Transactions on Graphics (TOG)* 24, 1 (2005), 140–164.
- Lin Shi and Yizhou Yu. 2005b. Taming liquids for rapidly changing targets. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*. ACM, 229–236.
- Jos Stam. 2002. A Simple Fluid Solver Based on the FFT. *J. Graph. Tools* 6, 2 (Sept. 2002), 43–52. <https://doi.org/10.1080/10867651.2001.10487540>
- Alexey Stomakhin and Andrew Selle. 2017. Fluxed Animated Boundary Method. *ACM Trans. Graph.* 36, 4, Article Article 68 (July 2017), 8 pages. <https://doi.org/10.1145/3072959.3073597>
- Nils Thuerey, Theodore Kim, and Tobias Pfaff. 2013. Turbulent Fluids. In *ACM SIGGRAPH 2013 Courses (SIGGRAPH '13)*. ACM, New York, NY, USA, Article 6, 1 pages. <https://doi.org/10.1145/2504435.2504441>
- Nils Thuerey and Tobias Pfaff. 2016. MantaFlow.(2016). URL <http://mantaflow.com> (2016).
- Nils Thuerey, Richard Keiser, Mark Pauly, and Ulrich Rüde. 2006. Detail-preserving fluid control. In *Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation*. Eurographics Association, 7–12.
- Adrien Treuille, Antoine McNamara, Zoran Popović, and Jos Stam. 2003. Keyframe control of smoke simulations. In *ACM Transactions on Graphics (TOG)*, Vol. 22. ACM, 716–723.
- Maximilian Werhahn, You Xie, Mengyu Chu, and Nils Thuerey. 2019. A Multi-Pass GAN for Fluid Flow Super-Resolution. *Proc. ACM Comput. Graph. Interact. Tech.* 2, 2, Article 10 (July 2019), 21 pages. <https://doi.org/10.1145/3340251>
- You Xie, Erik Franz, Mengyu Chu, and Nils Thuerey. 2018. tempoGAN: A Temporally Coherent, Volumetric GAN for Super-resolution Fluid Flow. *ACM Trans. Graph.* 37, 4, Article 95 (July 2018), 15 pages. <https://doi.org/10.1145/3197517.3201304>
- Jonas Zehnder, Rahul Narain, and Bernhard Thomaszewski. 2018. An Advection-Reflection Solver for Detail-Preserving Fluid Simulation. *ACM Trans. Graph.* 37, 4, Article 85 (July 2018), 8 pages. <https://doi.org/10.1145/3197517.3201324>
- Yongning Zhu and Robert Bridson. 2005. Animating Sand as a Fluid. *ACM Trans. Graph.* 24, 3 (July 2005), 965–972. <https://doi.org/10.1145/1073204.1073298>

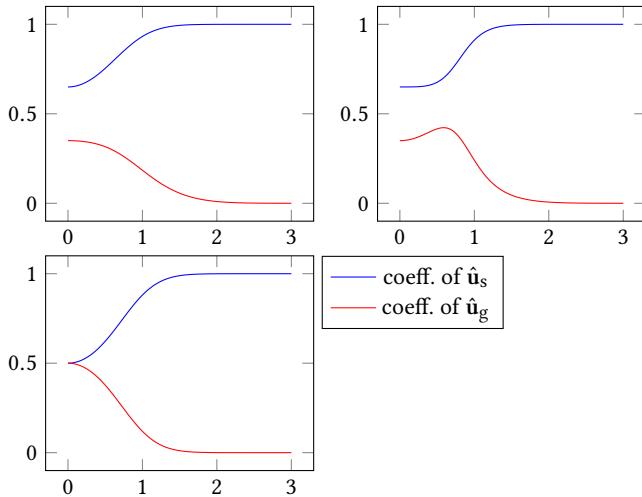


Fig. 11. Guiding coefficients of Nielsen et al. [2009] (top left), Nielsen and Christensen [2010] (top right), and Inglis et al. [2017] (bottom left) as a function of frequency  $v$ . We take  $L$  to be a Gaussian filter  $L(v) = e^{-v^2}$ , and set  $\alpha = 0.65$ ,  $\phi = 0.35$ , and  $W = 1$  as suggested by the respective authors.

## A FREQUENCY-DOMAIN ANALYSIS OF PREVIOUS WORK

Here we apply the frequency-domain analysis of Section 3.1 to the existing guiding methods of Inglis et al. [2017]; Nielsen and Christensen [2010]; Nielsen et al. [2009].

Let us first consider the method of Nielsen et al. [2009] with spatially uniform guiding (i.e. the guiding weight  $\alpha$  is independent of position). Translating their eq. (6) into our notation, their method finds the stationary point of

$$\iiint \left( \frac{\alpha}{2} \|\mathbf{u} - \mathbf{u}_s\|^2 - \lambda \nabla \cdot \mathbf{u} + \frac{1-\alpha}{2} \|L * \mathbf{u} - \mathbf{u}_g\|^2 \right) dV \quad (7)$$

with respect to  $\mathbf{u}$  and  $\lambda$ , where  $L$  is a low-pass filter kernel. This is equivalent to solving

$$\min_{\mathbf{u}} \iiint \left( \frac{\alpha}{2} \|\mathbf{u} - \mathbf{u}_s\|^2 + \frac{1-\alpha}{2} \|L * \mathbf{u} - \mathbf{u}_g\|^2 \right) dV \quad (8)$$

$$\text{s.t. } \nabla \cdot \mathbf{u} = 0. \quad (9)$$

We will ignore the divergence-free constraint  $\nabla \cdot \mathbf{u} = 0$ , since it turns out that when both  $\mathbf{u}_g$  and  $\mathbf{u}_s$  are divergence-free, the optimal  $\mathbf{u}$  is automatically divergence-free.

Parseval's theorem implies that  $\iiint \|\mathbf{u}\|^2 dV = \iiint \|\hat{\mathbf{u}}\|^2 d\hat{V}$ , so this is equivalent to minimizing

$$\iiint \left( \frac{\alpha}{2} \|\hat{\mathbf{u}} - \hat{\mathbf{u}}_s\|^2 + \frac{1-\alpha}{2} \|\hat{L}\hat{\mathbf{u}} - \hat{\mathbf{u}}_g\|^2 \right) d\hat{V}. \quad (10)$$

Since this function is separable over frequencies, we can independently minimize

$$\frac{\alpha}{2} \|\hat{\mathbf{u}} - \hat{\mathbf{u}}_s\|^2 + \frac{1-\alpha}{2} \|\hat{L}\hat{\mathbf{u}} - \hat{\mathbf{u}}_g\|^2 \quad (11)$$

at each frequency mode. This gives the optimal solution

$$\hat{\mathbf{u}} = \frac{\alpha}{\alpha + (1-\alpha)\hat{G}^2} \hat{\mathbf{u}}_s + \frac{(1-\alpha)\hat{G}}{\alpha + (1-\alpha)\hat{G}^2} \hat{\mathbf{u}}_g. \quad (12)$$

Thus, at each frequency mode, the guided velocity is a linear combination of the simulated velocity  $\hat{\mathbf{u}}_s$  and the guiding velocity  $\hat{\mathbf{u}}_g$  with frequency-dependent coefficients.

The method of Nielsen and Christensen [2010] solves

$$\min_{\mathbf{u}} \iiint \left( (1-\phi) \|L * \mathbf{u} - L * \mathbf{u}_s\|^2 + \phi \|L * \mathbf{u} - \mathbf{u}_g\|^2 + \|\mathbf{u} - L * \mathbf{u} - (\mathbf{u}_s - L * \mathbf{u}_s)\|^2 \right) dV = 0, \quad (13)$$

which by a similar argument yields

$$\hat{\mathbf{u}} = \frac{(2-\phi)\hat{L}^2 - 2\hat{L} + 1}{\hat{L}^2 + (1-\hat{L})^2} \hat{\mathbf{u}}_s + \frac{\phi\hat{L}}{\hat{L}^2 + (1-\hat{L})^2} \hat{\mathbf{u}}_g \quad (14)$$

for spatially uniform guiding.

Finally, Inglis et al. [2017] solve

$$\min_{\mathbf{u}} \iiint \left( \|L(\mathbf{u} - \mathbf{u}_g)\|^2 + \|W(\mathbf{u} - \mathbf{u}_s)\|^2 \right) dV, \quad (15)$$

yielding

$$\hat{\mathbf{u}} = \frac{W^2}{(\hat{L}^2 + W^2)} \hat{\mathbf{u}}_s + \frac{\hat{L}^2}{(\hat{L}^2 + W^2)} \hat{\mathbf{u}}_g. \quad (16)$$

The frequency-dependent coefficients for all three methods are visualized in Fig. 11.