EUROGRAPHICS 2021 / N. Mitra and I. Viola
(Guest Editors)

Volume 40 (2021), Number 2

Honey, I Shrunk the Domain: Frequency-aware Force Field Reduction for Efficient Fluids Optimization

Jingwei Tang[1], Vinicius C. Azevedo[1], Guillaume Cordonnier[1,2] and Barbara Solenthaler[1]

[1]ETH Zürich [2]Inria, Université Côte d'Azur

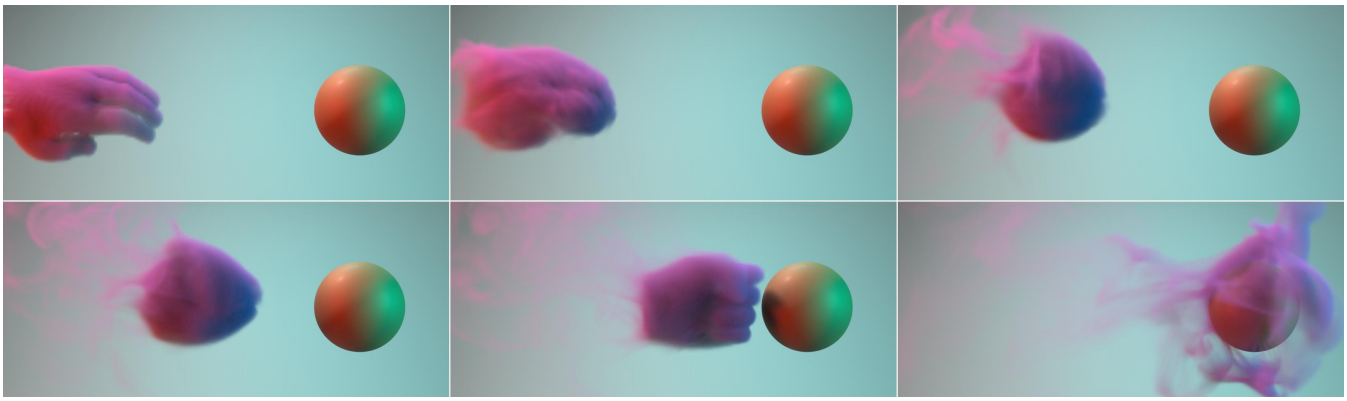


Figure 1: *Our method computes an incompressible force field to guide a smoke simulation, such that densities match a user-defined keyframe (pre-defined hand shape). The proposed reduced representation of the force field optimization uses vector potentials and progressive spectral filtering masks, enabling smooth transitions between the underlying physical phenomena and close matching with the target.*

Abstract

Fluid control often uses optimization of control forces that are added to a simulation at each time step, such that the final animation matches a single or multiple target density keyframes provided by an artist. The optimization problem is strongly under-constrained with a high-dimensional parameter space, and finding optimal solutions is challenging, especially for higher resolution simulations. In this paper, we propose two novel ideas that jointly tackle the lack of constraints and high dimensionality of the parameter space. We first consider the fact that optimized forces are allowed to have divergent modes during the optimization process. These divergent modes are not entirely projected out by the pressure solver step, manifesting as unphysical smoke sources that are explored by the optimizer to match a desired target. Thus, we reduce the space of the possible forces to the family of strictly divergence-free velocity fields, by optimizing directly for a vector potential. We synergistically combine this with a smoothness regularization based on a spectral decomposition of control force fields. Our method enforces lower frequencies of the force fields to be optimized first by filtering force frequencies in the Fourier domain. The mask-growing strategy is inspired by Kolmogorov's theory about scales of turbulence. We demonstrate improved results for 2D and 3D fluid control especially in higher-resolution settings, while eliminating the need for manual parameter tuning. We showcase various applications of our method, where the user effectively creates or edits smoke simulations.

CCS Concepts

• ***Computing methodologies*** → ***Physical simulation;***

1. Introduction

Fluid phenomena are ubiquitous, and their depiction in virtual environments is generally performed by evaluating the classical fluid equations of motion. As such, controlling automatically animated flow structures becomes an important tool in effects production: flow features have to be precisely modified to convey a dramatic effect or to aesthetically compose a shot. In production settings, the majority of fluid control tools and techniques are based on manual

and time-intensive adjustment of parameters and initial conditions. This tends to be non-optimal as it includes multiple trial-and-error iterations to find a good set of techniques and parameters that do not compromise the natural motion.

Thus, several methods aim to control fluid simulations by adding a set of forces at each time step, in an attempt to guide the fluid towards a specified objective. Differentiable flow solvers [TMPS03] and its popular adjoint method [MTPS04] extension, guide fluids by computing derivatives of the objective in terms of target density keyframes with respect to control parameters. These works inspired many modern fluid control optimization approaches. Variational dual grid approaches [NCZ*09], improving liquids optimizations [RTWT12], and the reformulation of the problem as a constrained optimization [IEGT17, PM17b] are different incarnations of the adjoint method in a differentiable setup.

However, actually implementing optimization-based control comes with challenges. Under-constrained objectives create transitions that are often unpleasant to the eye: as the animation progresses, structures may disappear or suddenly merge into the designed objective. Additionally, the dimension of the space of input controls coupled with expensive non-linear Navier-Stokes solvers still render the computational time of these methods impractically slow. Given that the gold standard for fluid control is to seamlessly bridge the gap between simulated fluid phenomena and artistically modified keyframes with affordable performance costs, current methods can still be improved.

Our work revisits this long-standing problem of fluid control when employing adjoint-based methods, in an attempt to overcome inherent limitations of previous work in terms of required parameter tuning, transition artifacts, and optimization performance. The core idea - simple, yet powerful - is to reduce the space of forces, and hence can be seen as a model reduction technique on control parameters. While it seems paradoxical that reducing the number of control variables enhances the quality and the visual aspect of obtained results, we empirically show that this is actually the case for fluid control. Our first insight relies on the fact that standard control methods allow the optimized forces to have divergent modes during the optimization process. These divergent modes are not entirely removed by the pressure solver step, manifesting as unphysical smoke sinks that can be explored by the optimizer to match a desired target. Thus, we reduce the space of the possible forces to the family of strictly divergence-free velocity fields by optimizing directly for a vector potential, which is discretely incompressible.

Our second insight is that at the beginning of the optimization, the optimizer struggles with the high frequency bands of the control force field. We hypothesize that due to the non-linear nature of the underlying equations, forces optimized locally will cascade down to turbulent structures as the simulation advances, especially in higher resolution scenarios. Hence, the optimization wastes computational cycles trying to revert back small structures by adding even finer structures, until it gets stuck on local minima. These high-frequency force fields produce structural deterioration that manifests into uncanny turbulent motion that does not resemble the natural phenomena. To tackle this issue, we propose a policy of first prioritizing the convergence of lower frequencies (bigger structures) of the force field. While this effectively reduces the search space in the beginning of the optimization, it can progressively converge to more intricate high-frequency patterns by adding more force field frequencies as the optimization progresses. Our policy could be embodied in many ways - our implementation specifically realizes this as relocating forces from the spatial to the Fourier domain, this way enabling the selection of specific bands in a way that mimics the natural Kolmogorov scale.

We synergistically combine these two space reduction techniques to obtain a novel and efficient method for fluid control, which substantially improves the quality of keyframe matching compared to previous approaches. We evaluate our method using various 2D and 3D examples and demonstrate the distinct advantage of our approach particularly for higher-resolution simulations.

2. Related Work

Unconstrained Optimization methods efficiently optimize general differentiable functions of the form

$$\min_{\mathbf{x}} \sum w_i f_i(\mathbf{x}), \tag{1}$$

where $\mathbf{x}$ are the input variables, are f_i objective functions with w_i correspondent weights. Unconstrained optimization is applied to a plethora of modern graphics applications ranging from mesh deformation [SCOL*04, SA07] and editing [BDS*12] to physics solvers [MTGG11, BML*14] and rendering [LADL18]. Gradient descent methods [Nes83, KB14] efficiently solve convex unconstrained optimized problems by computing the change of input parameters, which will locally minimize the target function the most. These methods are also widely used to train deep convolutional neural networks [KSH17], while working surprisingly well for non-convex functions. However, gradient-descent optimization might be inefficient, or even impracticable, when the Hessian is ill-conditioned. Thus, Quasi-Newton methods [FP63] approximate the Hessian matrix to compute accurate update directions. Among Hessian approximation approaches, the L-BFGS [ZBLN97] is a popular algorithm and it is widely used in several graphics applications [MTPS04, LBK17].

Constrained Optimization methods optimize objective functions of the form

$$\min_{\mathbf{x}} \sum w_i f_i(\mathbf{x}), \quad \textbf{s.t.}\ A\mathbf{x} = b, \tag{2}$$

in which we only consider linear constraints, since convex non-linear constraints can be incorporated by the use of indicator functions [FZB19]. Common methods to solve Equation (2) are the Alternating Direction Method of Multipliers (ADMM) [Boy10, ZPOD19], Douglas-Rachford Splitting [HH16, FZB19], Lagrange Multipliers [NCZ*09, TNGF15] and the Augmented Lagrangian Method [NSO12, ZZ18]. In graphics, ADMM has been employed to varies problems including geometry processing [BTP13, NVT*14], physics simulation [OBLN17] and control [GITH14, PM17b].

2.1. Controlling Fluid Simulations

Unconstrained Optimization for Fluids. Gradient descent algorithms require that the physics solver and all the components involved on the target optimized function to be at least first-order

differentiable. Computing derivatives with respect to model equations would produce a mismatch between the optimization and the advancement of the simulation states [PSE*00]; thus Treuille et al. [TMPS03] computed analytic derivatives relative to a fluid simulator. Their approach was the first to employ a non-linear optimization scheme to enable matching of target density keyframes by the use of control forces. Additionally, the authors coupled a L-BFGS algorithm with adaptive blurring kernels to improve convergence and reduce flat areas of the objective function. However, since the derivatives are computed in a *forward-pass*, each derivative computation of input forces requires a full solver evaluation, restricting the approach to very few control parameters.

Macnamara et al. [MTPS04] noticed that forward-pass derivatives could be computed in a dual-way (*backward-pass*) by the adjoint method, requiring only one solver evaluation for computing all necessary derivatives. The adjoint method was also employed to control liquids from sketches [PHT*13], where a non-linear Least Squares Levenberg-Marquardt solver is used employing a reduced set of control forces defined similarly as in Treuille et al. [TMPS03]. However, the adjoint method explicitly stores intermediate differentiable operations states, which might induce prohibitively expensive memory requirements for high-resolution simulations. Unconstrained optimization was also employed for animating pre-computed liquid sequences [RWTT14], interpolation between velocity fields from distinct simulations [SDN18], and optimizing graph-cuts for fluid carving [FEHM19].

Constrained Optimization for Fluids. Modern fluid control employs constrained optimizations by formulating the pressure projection step as a linear constraint to enforce incompressibility. Nielsen et al. [NCZ*09] was the first to explore this to guide high-resolution flows to match simulations from coarser resolutions. The proposed variational guiding does not require a non-linear optimization solve, since the control is embedded directly on the pressure projection step. This approach was further extended for better performance and memory consumption [NC10], supporting liquids guide-shapes and meshes [NB11, RTWT12], and manipulation of Lagrangian Coherent Structures [YCZ11]. Constrained control was also employed to recover fluid velocity and density fields from images [GITH14, EHT18], improving optimization efficiency [IEGT17, PM17b], interpolating liquids and smoke by space-time deformations [Thu16], improving boundary conditions handling [IEGT17], and video-guided parameter transfer for viscous fluids [TL19].

Feature-based Control. Beyond standard fluid control, some approaches rely on controlling fluids through feature manipulation. Ren et al. [RLL*13] modulate spectral information of flow fields by employing a Hilberg-Huang transform. Their approach allows incorporation of different smoke styles by suppressing or enhancing fluid frequencies. A library of 2-D liquid sketches was matched with artist inputs to artistically manipulate breaking waves [MMS04], while fluid sculpting of pre-computed simulations was computed interactively by matching space-time features [MVW*16]. Approaches based on vorticity formulations were employed to allow controlling features such as vortex filaments and rings [ANSN06, WP10]. Sato et al. [SDKN18] enhanced low resolution simulations with turbulence features extracted from high-resolution ones, while a transport-based neural style transfer extended this method for arbitrary image-based artistic manipulation [KAGS19].

Optimization-free Control. Alternative to full optimization, simpler approaches approximate forcing functions by evaluating guiding equations that are computed explicitly at each time-step, requiring little or no internal iterations. While these algorithms are popular due to their efficiency when compared with full optimization schemes, they do not guarantee that target shapes will be matched, hence the optimality of the solution is neglected. Fattal and Lichinski [FL04] derive forcing functions that are computed with closed form solutions to match user specified keyframes. Similar approaches were used to guide smoke by signed distance functions [SY05a, SDE05, YLYJ13], geometric potentials [HK04] and radial basis functions [PCS04], controlling liquids [REN*04, SY05b, TKPR09], matching high resolution simulations to lower resolution ones [HMK11], path [KMT06] and particle-based smoke control [MM13], and deforming simulations by warped grids [PM17a].

Differentiable Physics with Deep Learning. Differentiable physics solvers have recently been integrated into deep learning pipelines to make use of the automatic differentiation. Schenk and Fox [SF18] implemented a differentiable version of position-based fluids to enable robots to interact with liquids, and differentiable PDE solvers were used in conjunction with a hierarchical predictor-corrector scheme for 2D grid-based fluid problems [HTK20]. Differentiable solvers were also applied to rigid body mechanics [dABPSA*18, DHDw19], cloth [LLK19] and multiple materials [HAL*20].

3. Force-based fluid control

Fluid control in computer graphics aims to output a physically plausible fluid simulation given a set of user-friendly target parameters, matching as closely as possible a desired artistic intent. Our method restricts its discussion to simulations of incompressible smoke. We notice, however, that the analysis presented here is also viable for a broader set of fluid setups, including viscous and liquid simulations. We follow previous work [TMPS03, MTPS04] and use a sparse set of density keyframes as the primary control tool for artists. More specifically, we aim at finding a set of forces $\mathbf{f}$, which control the Navier-Stokes equations

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u}\cdot\nabla)\,\mathbf{u} = -\frac{1}{\rho}\nabla p + \mathbf{f}, \tag{3}$$

$$\nabla\cdot\mathbf{u} = 0, \tag{4}$$

where $\mathbf{u}$ and p are the fluid velocity and pressure fields. We omit viscosity terms due to the inherent dissipation of velocity-pressure fractional stepping methods [ETK*07, MCP*09]. At the end of a time-step, smoke densities ρ are passively advected by fluid velocities by $\frac{\partial \rho}{\partial t} = -\mathbf{u}\cdot\nabla\rho$.

We define the set of discrete forces over the interval $[i, j]$ as $\mathbf{f}_i^j = \{\mathbf{f}_i, \mathbf{f}_{i+1}, \ldots, \mathbf{f}_{j-1}, \mathbf{f}_j\}$. Allowing the force fields to vary while all other simulation variables are evaluated solely by Equation (3), the density at time t can be written as a $\rho_t = \phi(\mathbf{f}_0^t)$. Hence, the unconstrained optimization problem for matching a set of user-

specified keyframes ρ_k^* is formulated as

$$\mathbf{f}^* = \arg\min_f \sum_{k \in K} \|\gamma(\phi(\mathbf{f}_0^k)) - \gamma(\rho_k^*)\|_2^2, \tag{5}$$

where K is the set of keyframe numbers, and γ is an objective processing function that can aid convergence in flat regions of the energy landscape (i.e., blurring) and/or select certain regions of the objective to be matched (i.e., masking). Although the objective of Equation (5) is apparently quadratic, forces vary non-linearly due to the velocity transport term of Equation (3). The solution of this full optimization can be separated by decoupling the force fields over time. Pan et al. [PM17b] subdivided this problem into two constrained but independent optimizations. The authors can then explore the quadratic nature of the density matching objective, using a fixed-point iteration for higher efficiency.

The objective of Equation (5) is only sparsely defined in terms of input keyframes. Thus, the optimization problem becomes strongly under-constrained. This negatively influences generated in-between density fields, since they can arbitrarily vary and potentially produce non-physical results. The Tikhonov regularization term [GITH14] favors minimally energetic flows, while a total variation (TV) regularizer enforces continuity inside the fluid domain. These two regularization terms are commonly added by previous methods to the density matching objective, yielding

$$\mathbf{f}^* = \arg\min_f \sum_{k \in K} \|\gamma(\phi(\mathbf{f}_0^k)) - \gamma(\rho_k^*)\|_2^2 + \alpha \frac{1}{2} (\mathbf{f}_0^k)^T \mathbf{f}_0^k + \beta ||\nabla \mathbf{f}_0^k||_2^2. \tag{6}$$

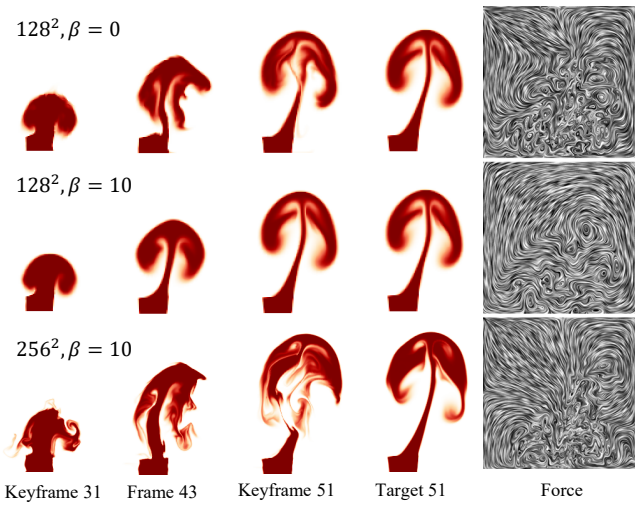
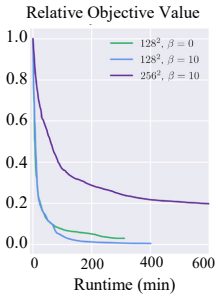


Figure 2: *Benchmark test using the baseline adjoint method [MTPS04] applied to a* uniform force *example for different resolutions (rows) using 3 keyframes as targets. The resulting force field is shown on the right.*

These regularization terms were successfully employed by several works [GITH14, EHT18]. However, one often overlooked aspect of Equation (6) is that finding proper weights (α and β) heavily biases the final solution, potentially inducing the convergence of the solution to a local minimum. To benchmark this full optimization problem, we devise a simple testing case with a directional (left to right) force that is constant in time applied to the velocity field. Hence, we fix the optimization variable to be a single force field $\mathbf{f}_0^T = \{\mathbf{f}_{gt}\}$ that does not vary in time, but can freely vary in space and is integrated to a 2D rising plume simulation. A gradient-based L-BFGS algorithm is employed to optimize forces with the same spatial dimensionality as the velocity field. This approach corresponds to the adjoint method implemented in [MTPS04].

Figure 2 shows the results of optimizing Equation (6) for three target keyframes with varying resolutions and smoothness weights. Our analysis excluded the Tikinov regularization ($\alpha = 0$), since it has a smaller effect than the TV regularization. The first row shows the converged result of the unweighted optimization ($\beta = 0$). Notice that while the objective was roughly matched at frames 31 and 51, the computed in-between (frame 43) suffers from unnatural motion. We empirically found that setting $\beta = 10$ effectively improves the quality of the results (second row). However, using this same regularization weight for a higher resolution simulation (256^2, third row) proved itself ineffective.



By analyzing the convergence of the objective function (inset plot), we conclude that the regularization weight option for the 256^2 resolution converged to a local minimum. This happens since the gradient-based optimization gets stuck in a solution visually far away from the target density objective, and is unable to further substantially decrease the objective function due to flat regions on the energy landscape. Another factor that plays a role to the convergence is the processing function γ. Varying degrees of Gaussian blur [TMPS03, MTPS04, IEGT17] and Gaussian pyramids [PM17b] have been used to aid the convergence. However, the selection of parameters used for blurring in these methods is still performed experimentally by the end user. The optimization of these parameters renders the optimization ineffective, since optimizing forces in this simple setup in 2D may take several hours.

4. Reduced-space forces for fluid control

As discussed in the previous section, fluid control methods suffer from the curse of dimensionality: the under-constrained nature of keyframe matching objective functions combined with the high-dimensionality of the force field pose challenges for finding optimal solutions. To remedy that, previous approaches conveniently wrote force fields in reduced parameter spaces, using a mapping function $\mathbf{f} = f(p)$, where p is a set of reduced parameters, to calculate forces on the spatial domain. For example, Treuille et al. [TMPS03] used Gaussian kernels to define a sparse set of wind and vortex forces that have smooth representation on the spatial domain. While these methods make the optimization more efficient, they trade-off accuracy for speed, since the optimality of the solution might be compromised by reducing the space of control forces indiscriminately. Additionally, adopting Gaussian kernels will involve the careful tuning of another set of parameters, which will further hinder the controllability of the method.

We propose two novel ideas that jointly tackle the lack of con-

straints and the high-dimensionality of force fields. Firstly, we noticed that divergent modes on force fields are (mostly) projected out by the incompressibility step. However, the optimization process is still able to create divergent forces that pass through the discrete pressure projection, manifesting as spurious sinks for intermediate density fields. This is a valid strategy found by the optimizer to match exactly the target keyframe, but it may lead to sub-optimal convergency. Thus, we propose to reduce the space of forces to the ones that are strictly divergence-free (up to discretization accuracy) by computing a mapping function as $\mathbf{f} = \nabla \times \psi$, where ψ is a vector potential function. This minimizes sinks on density in-betweens and leads to a physically plausible motion (Section 4.1).

Secondly, ensuring smooth solutions is paramount for generating pleasant and natural transitions between the underlying physical phenomena and the target keyframes. Thus, we propose a smoothness regularizer, in which lower frequencies of the force field are optimized first. We take inspiration from successful optimization of highly non-convex functions in Convolutional Neural Networks (CNNs), where first the lower frequencies of the error are approximated. Our optimizer is tailored to automatically detect when convergency becomes stale for a certain frequency band, increasing the number of included bands on the optimization to further reduce the approximation error. We will detail our spectrally-aware force field decomposition in Section 4.2. These two space reduction techniques combine synergistically to create density animations with higher quality than previous methods (Section 5).

4.1. Strictly Incompressible Force Fields

It is straightforward to replace the optimization variables $\mathbf{f}$ of Equation (6) by the vector potential ψ, which will generate strictly incompressible divergence-free force fields $\mathbf{f} = \nabla \times \psi$. We highlight that contrary to our strict divergence-free forces, divergent modes on force fields can still be explored by the optimizer. This happens since the pressure projection is discrete and has a limited accuracy, and increasing the residual accuracy of a conjugate gradient solver would induce additional overhead on the already expensive optimization pipeline. Thus, high divergence modes on optimized force fields can still pass through the projection step and create sinks, as shown for the 3D example in Figure 14.

While the curl operator is known to have a null-space in 3D which can be removed with a Couloumb gauge [ATW15] or a tree-cotree [AR90] regularization, we did not notice problems related to reduced convergence. We observed that these null-spaces are ignored by gradient-based optimizers: changes in the null-space direction yield the same underlying vector potential, which does not change the objective function. We notice that a tree-cotree regularization could potentially make the convergence of our vector potentials a bit faster, since the regularization removes about a third of the total optimized variables.

We show in Figure 3 how regularization affects results when optimizing Equation (6) with strictly incompressible 2D force field results. A higher smoothness weight is needed for the incompressible force field when compared to the baseline adjoint method (Figure 2). This happens because stream functions in 2D are inherently smoother than their velocity field counterparts. In Figure 15 we compare the strictly incompressible approach with the baseline adjoint method. The unconstrained force field creates sinks for in-between densities, which facilitates the convergence to the desired keyframe, at the cost of producing unnatural smoke transitions. These sinks are minimized for strictly incompressible forces.

4.2. Spectral Regularization of Force Fields

Finding a good balance between the target objective function and the smoothness regularization has proven to be a difficult task. Large weights tend to not converge to the desired target, while small ones might get stuck in local minima. We observed that the impact of the smoothness regularizer - and thus, the intricacy of tuning its weight - increases with the resolution. This could be explained by Kolmogorov's cascade theory [Ric22], which states that the fluids energy at low frequencies (big vortices) will cascade to higher frequencies (smaller vortices). Thus, forces added at the beginning of the simulation will evolve into smaller vortices as the simulation progresses, which yields a chaotic objective landscape that can be contaminated with high frequency information. This cascading effect is more pronounced in higher resolution grids, since they can model more frequency bands.

This observation motivates a spectral decomposition of force fields, where lower frequencies of the solution are optimized first by filtering forces in the Fourier domain. Once the low frequencies converge under a certain threshold, the number of covered spectral bands is increased by expanding a filter mask $\mathcal{M}$. It is not necessary to compute a Fourier transform of the original force field, since one can directly optimize for the Fourier coefficients by simply applying inverse transform $\psi = \mathcal{F}^{-1}(\Psi)$ and optimizing for Ψ. Combining the strictly divergence-free vector potentials with the spectral force decomposition, our method proposes the following optimization:

$$\Psi^* = \arg\min_{\Psi} \sum_{k \in K} \|\gamma(\phi(\nabla \times (\mathcal{F}^{-1}(\Psi \cdot \mathcal{M})) - \gamma(\rho_k^*)\|_2^2. \quad (7)$$

Notice that the above equation has no regularization terms, avoiding the reliance on manually optimized weighting (α, β in Equation (6)). The only parameter is the mask $\mathcal{M}$, since it controls which frequencies are optimized at a given optimization phase. We propose to grow the mask when the optimization has reached a given tolerance, i.e., , when $|\mathcal{L}_{i+1}(\Psi) - \mathcal{L}_i(\Psi)|/\mathcal{L}_i(\Psi) < \tau$, where $\mathcal{L}_i$ is the objective function value at iteration i and τ is a user-defined threshold.

Figure 3: *Vector potential results illustrating how the weight of TV regularization affects the optimization result.*

4.2.1. Kolmogorov-inspired Priors

There are several strategies to grow the mask $\mathcal{M}$ once it satisfies a defined threshold. For example, its size could grow as a polynomial or exponential function. We chose a physically-inspired exponential mask-growing strategy based on Kolmogorov theory [FK95]. The Kolmogorov theory describes how vortex energy cascades to smaller scales, down to frequencies that are dominated by dissipation. For a given wavenumber k within the inertial subrange, the energy spectrum of turbulence $E(k)$ gives the amount of turbulence kinetic energy in an infinitesimal neighborhood around k as

$$E(k) = C\varepsilon^{2/3}k^{-5/3}, \tag{8}$$

where C is the Kolmogorov constant, and ε is the dissipation rate. Turbulence synthesis methods [SF93, KTJG08] commonly employ Kolmogorov theory to control the energy spectrum of smoke simulations. The total energy of the frequency bands above wavenumber k is given by integrating Equation (9):

$$E(k > k_0) = \int_{k_0}^{\infty} C\varepsilon^{2/3}k^{-5/3}\,\mathrm{d}k = \frac{2}{3}C\varepsilon^{2/3}k_0^{-2/3}. \tag{9}$$

Let us assume now that the inertial subrange starts at a wavenumber k_0, and we want to split the frequency domain in two parts, each of these having the same total energy. Then, the splitting wavenumber k_1 verifies that $E(k > k_0) = 2E(k > k_1)$, and from Equation (9) we obtain $k_1 = 2^{3/2}k_0$. This splitting is applied recursively to obtain a series of frequencies that can ideally filter forces as:

$$k_n = 2^{3n/2}k_0. \tag{10}$$

Our filters are designed to be ideal low-pass filters in the Fourier frequency domain, which enables fine-grained control on the frequency cutoff. In this way, the mask $\mathcal{M}$ suppresses all forces in the frequency domain Ψ whose L^∞ norm is above the cutoff wavenumber k_n

$$\mathcal{M} = \begin{cases} 1, \text{ if } \|\Psi\|^\infty \leq k_n, \\ 0, \text{ otherwise.} \end{cases} \tag{11}$$

which generates the rectangular mask shape shown in Figure 4.$k_0 = 1$ is chosen for our experiments. We notice that ideal band-pass filters might induce ringing artifacts. However, this seemed to not be an issue for the several examples in which we tested our approach (Section 5).

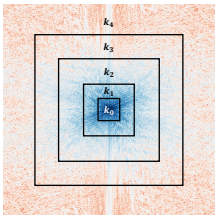


Figure 4: *Schematic mask growing strategy*

5. Results

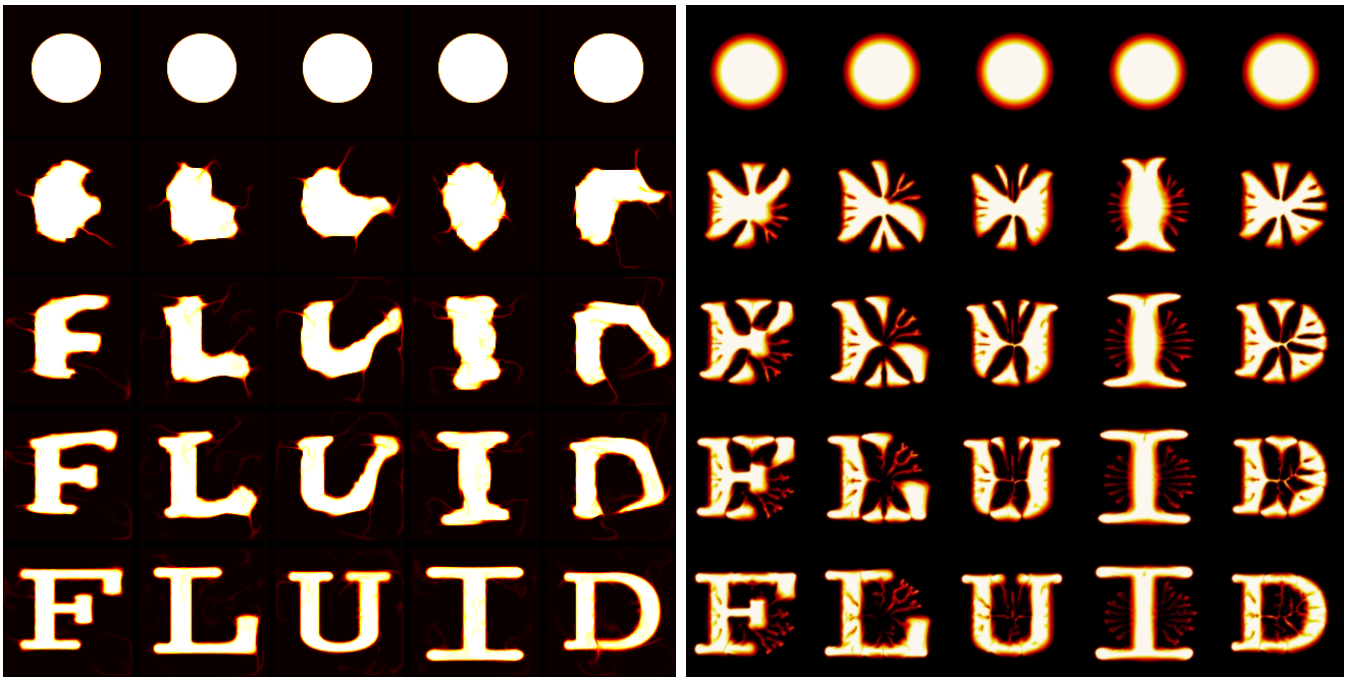
We designed several experiments to compare our results with previous work and explore several smoke editing applications. Our method can handle multiple keyframes as input, and the presented results use between 1 and 6 keyframes For many of the presented results, we define the *Baseline* to be the adjoint method [MTPS04] (Equation (5)), and *Baseline w/* TV the adjoint method with smoothness regularization (Equation (6)) and *ours* as the Kolmogorov-inspired mask growing method with strictly divergence free force fields (Equation (7)). The work of [PM17b] provides a method that is complementary to ours, since it splits the solution into two objectives that are not equivalent to directly optimizing for Equation (5). Thus, we only provide a single comparison between results taken from their paper and our approach in Figure 5. Our accompanying video shows animated sequences for the results presented in this section.

5.1. Implementation

Our fluid simulation solver and the optimization algorithm are implemented on Python with PyTorch [PGM*19], since this framework offers an embedded automatic differentiation tool as well as easy GPU deployment. We run all our 2D experiments on an NVIDIA GeForce RTX 2080Ti GPU with 11 GB of dedicated memory, and our 3D experiments on NVIDIA Tesla V100-SXM2 GPU with 32 GB dedicated memory.

Simulation. Density, pressure, velocity and vector potential fields are discretized on an orthogonal and equidistant staggered MAC grid [HW65]. Specifically, density and pressure values are located at cell centers, while velocities are stored on the faces of cells, and vector potential values are stored on the edges. The Semi-Lagrangian method is employed to solve the advection step with an 3rd order Runge-Kutta integrator, except for examples in Figure 14 (left), Figure 15 (left) and Figure 13, where a 1st order Euler integrator is used. To enforce the incompressibility of the fluid, we implement pressure projection through a Preconditioned Conjugate Gradient (PCG) solver, with Incomplete Poisson Preconditioner [SKF11], which can be easily parallelized. The tolerance of the PCG solver is set to be 0.001.

Optimization. Equation (7) is solved as an unconstrained optimization problem with a Quasi-Newton (L-BFGS) optimizer. We choose a history size of 10 to approximate the Hessian, and the Weak Wolfe criteria is used for the line search algorithm, which is limited to 25 steps. For all tests the same hyper-parameters are used for the L-BFGS algorithm. All keyframes are optimized together at once in all experiments except the one shown in Figure 13. The per-frequency tolerance for the optimization is optionally set by the user, and we choose the tolerance as either $\tau = 5 \cdot 10^{-5}$ for 2-D examples or $\tau = 1 \cdot 10^{-5}$ for 3-D examples. Since, the L-BFGS optimizer is based on gradient descent, the gradient of the objective function is computed at each optimization step. We employ a fully automatized adjoint method [MTPS04] as the fluid solver (Equation (3)) is implemented with automatic differentiation embedded in PyTorch. This makes the implementation of the optimization method fairly straightforward, with an exception of the pressure projection. A naive implementation of any iterative method (e.g., conjugate gradient) to solve this step using automatic differentiation would require the backpropagation to go through all the iterations, resulting in an significant memory bottleneck. We shortcut the automatic differentiation by directly computing the adjoint of the pressure projection, following [MTPS04]. Another advantage



Ours *[PM17b]*

Figure 5: *Our method (left) is compared against the method of [PM17b] (right) on FLUID example at resolution* 128^2. *From the top row to the bottom row, we show the results at frames 0, 20, 30, 35, 40 respectively.*

of implementing the fluid solver with an automatic-differentiation pipeline is that it allows us to easily plug and modify various components in the optimization process. In particular, there is no need to analytically compute the gradients of the inverse Fourier transform.

5.2. 2-D Test cases

We first show simple test cases to better illustrate differences of our method when compared to a standard adjoint approach. In this section, all 2-D target simulations are assumed to have a single force field. A sparse set of keyframes from these simulations are selected as target densities for the optimization in order to evaluate the ability of various different approaches to reach these keyframes. Extensive comparisons for additional 2-D cases examples are provided in the supplementary material.

Impact of Resolution. Our first experiment shows the impact of the simulation resolution, and thus of the parameter space, on the convergence of the optimization. For that, we use the same benchmark test case in Section 3, where a uniform force field pushes a buoyant smoke plume towards the right. As shown in Figure 6, the optimization converges well for a low resolution (64^2) for both Baseline and Baseline w/ TV methods. But the quality of the two methods degrades when the resolution increases (128^2, then 256^2), while our results at the keyframe remain indistinguishable from the target, regardless of the resolution.

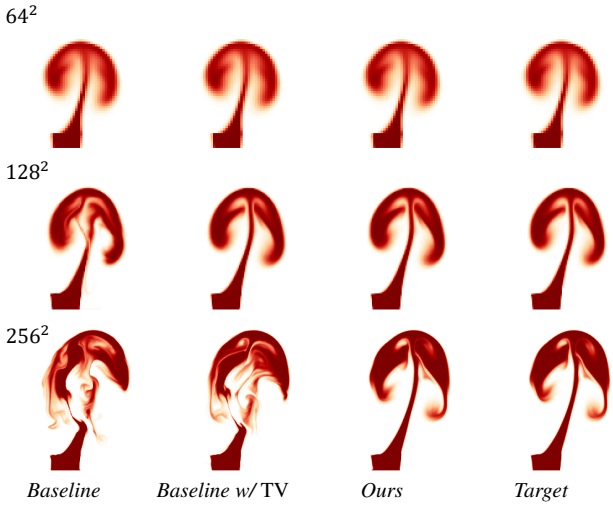


Figure 6: *Uniform force example for different simulation resolutions (rows) and optimization strategies (columns). 3 keyframes are used as target while only the last one is shown.*

5.2.1. Multi-scale vortices example

In this example, a set of Gaussians scalar fields with varying amplitude and variance are used as stream functions to generate a complex force profile (Figure 7, bottom right). Figure 7 also shows density and force fields obtained when stopping the algorithm after the convergence of specific wavenumbers $k_c = 1, 3, 64$. While

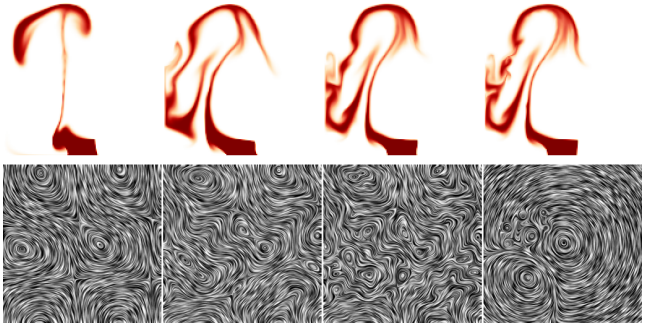


Figure 7: *Multi-scale vortices example at resolution* 128^2. *Density results (top row) and corresponding line integral convolution of force fields (bottom row), obtained by our method when filtering up to a selected wavenumber* k_c.

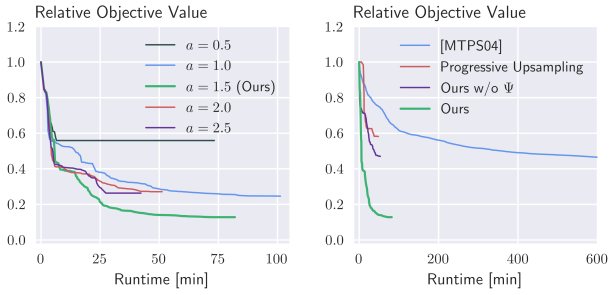


Figure 8: *Convergence plot on* Multi-scale vortex force *example at resolution* 256^2. *The plot on the left shows different mask growing strategies by varying the constant a in* $k_n = 2^{an}k_0$. *Our method with Kolmogorov-inspired mask growing strategy (green) converges much faster and to a better minimum than other strategies. The plot on the right compares our methods (green) with other optimization strategies: Progressive Upsampling (red), Ours w/o* Ψ *(purple) and Baseline w/* TV *(blue).*

higher wavenumbers are necessary for the simulation to fully reach the target, lower ones enforce smoother, less turbulent transitions. Lastly, to demonstrate that our Kolmogorov-inspired weights are better than arbitrary exponential weights, we show a convergence plot (Figure 8, left) for varying exponential values constants (p). Our approach (green) shows a better convergence profile.

Additionally, we implemented a progressive upsampling of vector potentials in the spatial domain as an alternative reduced domain strategy that does not rely on Fourier decomposition, but follows the same policy of converging lower frequencies first (Figure 9). Progressive upsampling is implemented as follows: U is the operator which trilinearly interpolates a volume of size n^3 to a volume of size $(2(n-1)+1)^3$, where the values are stored at the center of a cell. We define a sequence of progressively growing $\{\psi_i\}_{i\in[1,N]}$, where ψ_i is of size 2^i+1, and ψ_N covers the whole simulation volume. Starting from an initial guess $\psi_1 = 0$, we iteratively solve

$$\arg\min_{\psi_i} \sum_{k\in K} \|\gamma(\phi(\nabla\times(U^{N-i}(\psi_i)))) - \gamma(\rho_k^*)\|_2^2$$

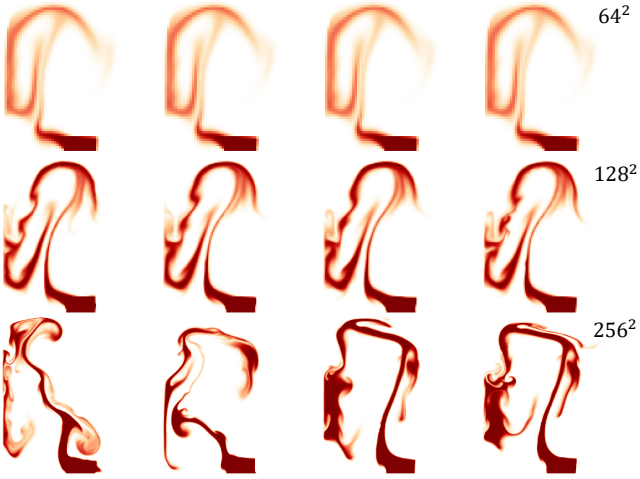


Figure 9: *Multi-scale vortex force example for different simulation resolutions (rows) and optimization strategies (columns). 6 keyframes are used in this example while only the last frame is shown here.*

and update the next initial guess by $\psi_{i+1} = U(\psi_i)$. The Kolmogorov-inspired mask growing in frequency domain achieves better results in terms of final objective value and visual effect than progressive upsampling, and converges much faster as shown in Figure 8. Both reduced domain strategies show better convergence and visual results than the method of Baseline and Baseline w/ TV. Moreover, we found that progressive upsampling tends to get into a bad local minima more easily than the Kolmogorov approach (Figure 9, first column). We also evaluated the impact of removing the strict divergence free constraint on forces (Figure 9, second column), while keeping our lower frequencies first policy. It is clear that using strictly incompressible forces converge to a better result (Figure 8, right).

5.3. Applications

We evaluate the usability of our method by applying it to several 3D examples of fluid control that mimic typical use cases of digital artists. The performance of all our examples is shown in Table (1), where we compare the time needed to reach convergence between Baseline w/ TV and our method. It can be seen that our approach outperforms the baselines in all our examples, and depending on the example, we reach a speed-up factor of up to 9×. It is notable that our band-growing approach offers the advantage that previews can be provided already after low bands have converged. This property is especially useful for prototyping, and further discussed below in the paragraph *Low Frequency Preview*. The maximal resolution that we used in our tests is 256^2 in 2D and 128^3 in 3D, which has matching resolutions with state-of-the-art control methods [PM17b].

Deforming Existing Simulations. One way of controlling a fluid is to start from an existing simulation, and then use standard mod-

eling tools to deform it. The user needs to deform only a few keyframes and to provide these as a target to our method. In Figure 10, the user twisted a vertical smoke plume in different directions at two keyframes. The baseline managed to closely reach the keyframe, but at the cost of a very turbulent simulation and spurious density sources and sinks. The smoothness regularization added to the baseline is very efficient at smoothing out the turbulence, but fails to reach the keyframes. On the contrary, our method smoothly reaches the target without the need of parameter tuning.

Figure 10: *3D twist example at resolution* $100 \times 200 \times 100$. *The method of Baseline (top row) does not precisely match target keyframes (bottom row). Our method (middle row) generates smooth transitions and matches target keyframe well.*

Matching Predefined Shapes. In Figure 11, we initialize the first density with a dragon-shaped volume, and we specify as a target keyframe the letters of EG2021, scaled so that the total volume of both shapes is identical. Comparisons are again shown for all optimization variants, highlighting the effectiveness of our approach to optimally matching the target and reducing spurious artifacts in in-betweens. In a different setting, we use the simplified differentiable renderer proposed by [KAT*19] to control a simulation with 2D strokes. We designed an initial simulation (Figure 12, top row), visualize it with the simplified renderer (inset image at top row), allowing the user to paint on the resulting rendering. The user can paint strokes to add and remove densities, and use image warping tools to deform the densities locally (inset image at bottom row). We use this renderer as the transformation function γ in our objective function (Equation (7)), replacing $\gamma(\rho^*)$ by the edited image. The differentiability of the simplified renderer allows us to compute the gradient of the objective function w.r.t to γ, making it compatible with our gradient-based optimization method.

Solid Obstacles. Our method is easily extended to include solid obstacles, because boundary conditions do not affect the gradients with respect to the force field. The force values lying inside the solid obstacles also do not affect the simulation, because the pressure project step can handle the velocity field penetrating into solid obstacles. We show the result in Figure 13, where target keyframes are defined in a scene with a solid sphere. The keyframe is defined every 30 frames. Due to memory limitations, we optimize for each keyframe separately, by using the density and velocity fields at the final time step of the last optimization as initial density and velocity field. Our approach successfully matches the target keyframes without apparent discontinuities in-between keyframes.

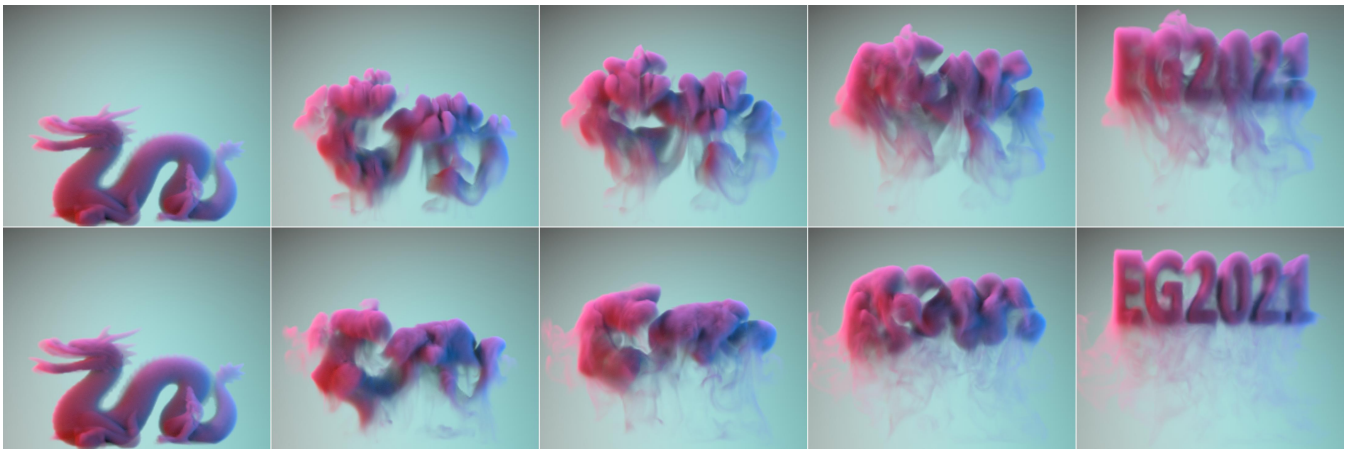
Density Sinks. The *3D uniform force* scene computed at resolution 128^3 shown in Figure 14 illustrates the problem of density sinks of Baseline when using 1^{st} order Euler integrator for advection. High divergence modes of the optimized force field can pass through the projection step and create sinks for the in-between densities that allow the method to converge to the keyframes. Dissipation of smoke is prevented in our approach due to the strictly incompressible force field, and better transitions are obtained. It can be seen that Baseline produces unwanted sinks and sources, leading to deviations from the ground truth density. A *Rubber toy* mesh is used as the target keyframe in Figure 15. The top row shows again noticeable smoke sinks when using Baseline, which are created in an attempt to match the target keyframe. The method of Baseline w/ TV shown in the middle row prevents density sinks, but fails to match the target keyframe. Our method in the third row can match the keyframe well with less density dissipation in in-between frames.

Low Frequency Preview. Force fields including only lower frequency bands require less computational time, and can serve as a preview of the final result, which is especially useful for authoring simulations. This advantageous property is illustrated in Figure 16. Once higher frequency bands are added, results display localized details while also preserving the main trajectory of the smoke. From the top row to the bottom row of Figure 16, corresponding cutoff wavenumber k's are 1, 8, 64. The runtime for each result is 166, 500 and 824 minutes respectively.

6. Discussion and Conclusion

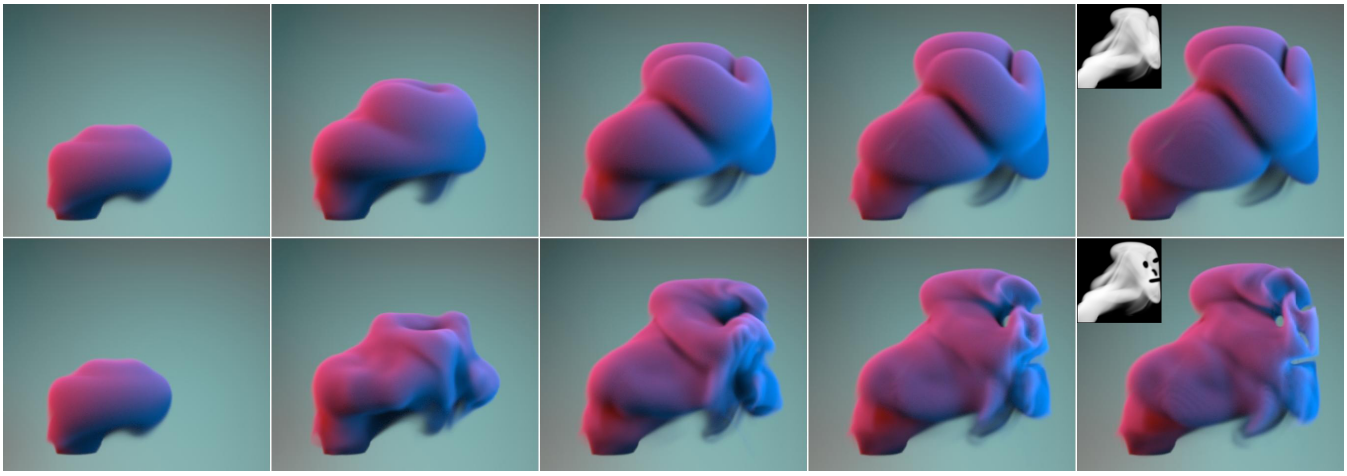
We have presented a simple, yet powerful technique to address the lack of constraints and high-dimensionality of the parameter space. Our method advances previous work on fluid control by providing better error convergence and density preservation, smoother transitions between simulations and keyframes, while also removing the need of hyper-parameter tuning of regularization weights. Another key property is that our method reliably converges also for high-resolution simulations, where previous optimization variants failed or produced artifacts in form of spurious density variations.

One of our main contributions is the policy of converging low frequencies first. The Kolmogorov-inspired masking strategy used on the frequency domain are a way to embody this idea into a concrete implementation and our method removes the need of traditional smoothness regularization such as TV. Additionally, the spectral regularization of force fields could be implemented in several ways, using for example wavelets, discrete cosine transforms,



Frame 0 *Frame 15* *Frame 20* *Frame 25* *Keyframe 30*

Figure 11: *3D dragon example at resolution* $200 \times 170 \times 60$*. The density is first initialized with a dragon-shaped volume, and the letters of EG2021 are specified as target. The method of Baseline (top row) fails to match the letters while our method (bottom row) matches target better.*



Frame 0 *Frame 15* *Frame 25* *Frame 28* *Keyframe 30*

Figure 12: *3D ghost example at resolution* $128 \times 128 \times 128$*. We couple our method with a simplified differentiable renderer to control a simulation with 2D strokes. Row 1: original simulation, Row 2: ours. The user edits the simplified rendered image (Row 1, inset) to create a target image (Row 2, inset)*

multi-level schemes, or varying smoothness weights as optimization advances. We choose using masking forces on the Fourier domain because they allow us a fine-grained control over the optimized frequencies, enabling the Kolmogorov-based mask growing approach. We chose to use the *Pytorch* framework for our implementation, which lacks support for sparse Fourier transforms. Using such a sparse representation might greatly improve the memory consumption of our method. Additionally, it might be possible to work on a coarser level and upsample it to a fine resolution. This would require its analytic derivatives and that frequencies are unchanged by cubic interpolation.

The problem of vanishing simulation densities in fluid control was never accounted for before. We believe that it is a novel finding that it is important to have divergence-free force fields, since this prevents density fields from vanishing during the simulation. Using the curl of the vector potential ensures force fields that are strictly divergence-free up to second-order accuracy of the employed operator discretization. Employing a pressure solver that is very precise to counteract divergent modes might reduce the problem but make fluid control methods inefficient. In our experiments, reducing the residual of the pressure solver for computing optimization forces easily takes our run time from 3-4 hours (10^{-3} tolerance) to 24 hours (10^{-6} tolerance). Therefore, we follow previous work [IEGT17] and use pressure solvers with higher residual tolerance. Increasing the tolerance leads to less accurate velocities and corresponding gradients, which could potentially affect the convergence of the optimization and result in sub-optimal solutions. In our experiments we found, however, that a high tolerance (10^{-3}) converges to similar results as a lower one (10^{-6}), both in quality and

Figure 13: *3D example with a solid obstacle at resolution* 128^3. *Keyframes are defined every 30 frames. Due to memory constraints, we use a first order Euler integrator for this example, and the optimization is conducted separately for each keyframe. Our method (top row) can match the targets (bottom row) well without obvious discontinuities.*

Table 1: *Performance comparison between methods of Baseline w/* TV *and Ours for the various example scenes. The number of minutes show the total amount of time needed by the optimizer before converging to a minimum.*

Scene	Resolution	# Frames	# Keyframes	Baseline w/ TV [min]	Ours [min]
2D uniform force (Fig. 6)	64×64	51	3	88	36
2D uniform force (Fig. 6)	128×128	51	3	406	67
2D uniform force (Fig. 6)	256×256	51	3	1387	211
2D multi-scale vortex (Fig. 7)	64×64	61	6	187	146
2D multi-scale vortex (Fig. 7)	128×128	61	6	329	176
2D multi-scale vortex (Fig. 7)	256×256	61	6	754	86
3D uniform force (Fig. 14)	$128 \times 128 \times 128$	31	1	1009	551
3D toy (Fig. 15)	$128 \times 128 \times 128$	30	1	970	677
3D twist (Fig. 10)	$100 \times 200 \times 100$	31	3	957	483
3D dragon (Fig. 11)	$200 \times 170 \times 60$	30	1	1440	793
3D ghost (Fig. 12)	$128 \times 128 \times 128$	30	1	1440	348
3D obstacle (Fig. 13)	$128 \times 128 \times 128$	90	3	N/A	583
3D soft punch (Fig. 16)	$200 \times 100 \times 100$	30	1	1440	824

number of iterations. Lastly, the choice of the position integrator of the advection step can also drastically influence density dissipation, as shown in Figure 15, in which our strictly divergence-free force fields also help.

7. Acknowledgments

This work was supported by ETH Zurich Research Grants (Grant No. ETH-08 18-1).

References

[ANSN06] ANGELIDIS A., NEYRET F., SINGH K., NOWROUZEZAHRAI D.: A Controllable, Fast and Stable Basis for Vortex Based Smoke Simulation. In *Proceedings of the 2006 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Aire-la-Ville, Switzerland, Switzerland, 2006), SCA '06, Eurographics Association, pp. 25–32. URL: http://dl.acm.org/citation.cfm?id=1218064.1218068. 3

[AR90] ALBANESE R., RUBINACCI G.: Magnetostatic field computations in terms of two-component vector potentials. *International Journal for Numerical Methods in Engineering 29*, 3 (mar 1990), 515–532. URL: http://doi.wiley.com/10.1002/nme.1620290305, doi:10.1002/nme.1620290305. 5

[ATW15] ANDO R., THUEREY N., WOJTAN C.: A stream function solver for liquid simulations. *ACM Trans. Graph. 34*, 4 (July 2015). URL: https://doi.org/10.1145/2766935, doi:10.1145/2766935. 5

[BDS*12] BOUAZIZ S., DEUSS M., SCHWARTZBURG Y., WEISE T., PAULY M.: Shape-Up: Shaping Discrete Geometry with Projections. *Computer Graphics Forum 31*, 5 (aug 2012), 1657–1667. URL: http://doi.wiley.com/10.1111/j.1467-8659.2012.03171.x, doi:10.1111/j.1467-8659.2012.03171.x. 2

[BML*14] BOUAZIZ S., MARTIN S., LIU T., KAVAN L., PAULY M.:

Figure 14: *3D uniform force example at resolution* 128^3 *using methods of Baseline (top row) and Ours (bottom row). The method of Baseline creates density sinks to match the target keyframe by exploiting the numerical dissipation of a first order Euler integrator (left three columns), but fails to match the target keyframe when* 3^{rd} *order Runge-Kutta integrator is used (right three columns). Our method generates smooth transitions and matches the keyframe target well under both integrator schemes.*

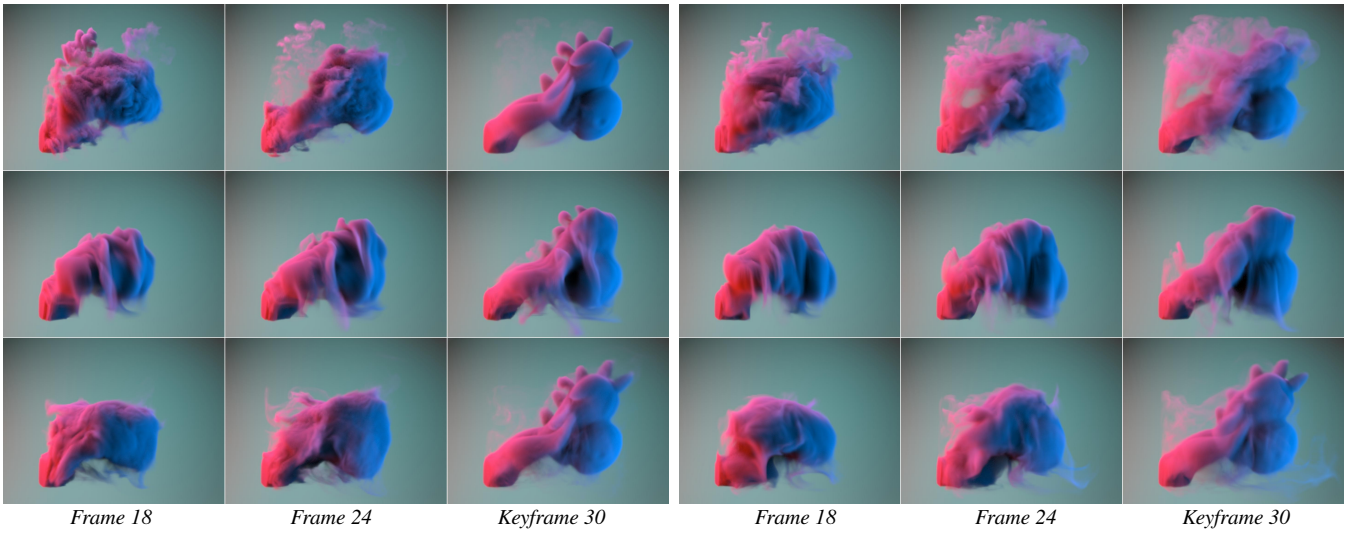
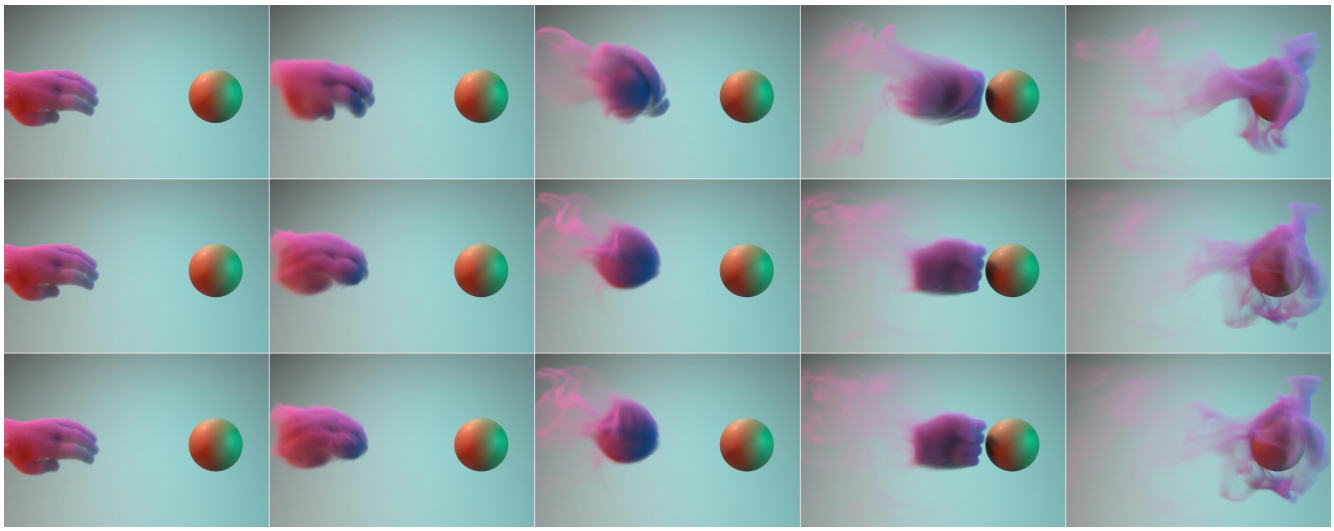


Figure 15: *3D toy example at resolution* 128^3 *using methods of Baseline (top row), Baseline w/* TV *(middle row) and Ours (bottom row). A pre-defined rubber toy shape is used as a target. The method of Baseline creates density sinks to match the target keyframe by exploiting the numerical dissipation of Euler integrator (left three columns), but fails to match the target keyframe when* 3^{rd} *order Runge-Kutta integrator is used (right three columns). The method of Baseline w/* TV *prevents the density sinks but fail to match the target keyframe. Our method generates smooth transitions and matches the keyframe target well under both integrator schemes.*

Projective dynamics. *ACM Transactions on Graphics 33*, 4 (jul 2014), 1–11. URL: http://dl.acm.org/citation.cfm?doid=2601097.2601116, doi:10.1145/2601097.2601116. 2

[Boy10] BOYD S.: Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers. *Foundations and Trends® in Machine Learning 3*, 1 (2010), 1–122. URL: http://www.nowpublishers.com/article/Details/MAL-016, doi:10.1561/2200000016. 2



Frame 0 *Frame 10* *Frame 20* *Keyframe 30* *Frame 40*

Figure 16: *3D soft punch example with a solid obstacle at resolution* $200 \times 100 \times 100$. *Two pre-defined hand shapes are used as initial frame and target frame. Our optimization results at different frequency bands are shown. From the top to the bottom, the cutoff wavenumber* k_c *is 1, 8, 64 respectively.*

[BTP13] BOUAZIZ S., TAGLIASACCHI A., PAULY M.: Sparse Iterative Closest Point. *Computer Graphics Forum 32*, 5 (aug 2013), 113–123. URL: http://doi.wiley.com/10.1111/cgf.12178, doi:10.1111/cgf.12178. 2

[dABPSA*18] DE AVILA BELBUTE-PERES F., SMITH K., ALLEN K., TENENBAUM J., KOLTER J. Z.: End-to-end differentiable physics for learning and control. In *Advances in Neural Information Processing Systems 31*, Bengio S., Wallach H., Larochelle H., Grauman K., Cesa-Bianchi N., Garnett R., (Eds.). Curran Associates, Inc., 2018, pp. 7178–7189. 3

[DHDw19] DEGRAVE J., HERMANS M., DAMBRE J., WYFFELS F.: A differentiable physics engine for deep learning in robotics. *Frontiers in Neurorobotics 13* (2019), 6. doi:10.3389/fnbot.2019.00006. 3

[EHT18] ECKERT M.-L., HEIDRICH W., THUEREY N.: Coupled Fluid Density and Motion from Single Views. *Computer Graphics Forum 37*, 8 (dec 2018), 47–58. URL: http://doi.wiley.com/10.1111/cgf.13511, doi:10.1111/cgf.13511. 3, 4

[ETK*07] ELCOTT S., TONG Y., KANSO E., SCHRÖDER P., DESBRUN M.: Stable, circulation-preserving, simplicial fluids. *ACM Trans. Graph. 26*, 1 (Jan. 2007), 4–es. URL: https://doi.org/10.1145/1189762.1189766, doi:10.1145/1189762.1189766. 3

[FEHM19] FLYNN S., EGBERT P., HOLLADAY S., MORSE B.: Fluid carving:intelligent resizing for fluid simulation data. *ACM Transactions on Graphics 38*, 6 (2019), 1–14. URL: http://dl.acm.org/citation.cfm?doid=3355089.3356572, doi:10.1145/3355089.3356572. 3

[FK95] FRISCH U., KOLMOGOROV A. N.: *Turbulence: the legacy of AN Kolmogorov*. Cambridge university press, 1995. 6

[FL04] FATTAL R., LISCHINSKI D.: Target-driven smoke animation. In *ACM SIGGRAPH 2004 Papers on - SIGGRAPH '04* (New York, New York, USA, 2004), ACM Press, p. 441. URL: http://portal.acm.org/citation.cfm?doid=1186562.1015743, doi:10.1145/1186562.1015743. 3

[FP63] FLETCHER R., POWELL M. J. D.: A Rapidly Convergent Descent Method for Minimization. *The Computer Journal 6*, 2 (aug 1963), 163–168. URL: https://academic.oup.com/comjnl/article-lookup/doi/10.1093/comjnl/6.2.163. 2

[FZB19] FU A., ZHANG J., BOYD S.: Anderson Accelerated Douglas-Rachford Splitting. URL: http://arxiv.org/abs/1908.11482, arXiv:1908.11482. 2

[GITH14] GREGSON J., IHRKE I., THUEREY N., HEIDRICH W.: From capture to simulation. *ACM Transactions on Graphics 33*, 4 (jul 2014), 1–11. URL: http://dl.acm.org/citation.cfm?doid=2601097.2601147, doi:10.1145/2601097.2601147. 2, 3, 4

[HAL*20] HU Y., ANDERSON L., LI T.-M., SUN Q., CARR N., RAGAN-KELLEY J., DURAND F.: Differentiable programming for physical simulation. In *International Conference on Learning Representations* (2020). URL: https://openreview.net/forum?id=B1eB5xSFvr. 3

[HH16] HE H., HAN D.: A distributed Douglas-Rachford splitting method for multi-block convex minimization problems. *Advances in Computational Mathematics 42*, 1 (feb 2016), 27–53. URL: http://link.springer.com/10.1007/s10444-015-9408-1, doi:10.1007/s10444-015-9408-1. 2

[HK04] HONG J.-M., KIM C.-H.: Controlling fluid animation with geometric potential. *Computer Animation and Virtual Worlds 15*, 34 (jul 2004), 147–157. URL: http://doi.wiley.com/10.1002/cav.17, doi:10.1002/cav.17. 3

[HMK11] HUANG R., MELEK Z., KEYSER J.: Preview-based sampling for controlling gaseous simulations. In *Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation - SCA '11* (New York, New York, USA, 2011), ACM Press, p. 177. URL: http://dl.acm.org/citation.cfm?doid=2019406.2019430, doi:10.1145/2019406.2019430. 3

[HTK20] HOLL P., THUEREY N., KOLTUN V.: Learning to control {pde}s with differentiable physics. In *International Conference on Learning Representations* (2020). URL: https://openreview.net/forum?id=HyeSin4FPB. 3

[HW65] HARLOW F. H., WELCH J. E.: Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. *Physics of Fluids 8*, 12 (1965), 2182–2189. doi:10.1063/1.1761178. 6

[IEGT17] INGLIS T., ECKERT M.-L., GREGSON J., THUEREY N.: Primal-Dual Optimization for Fluids. *Computer Graphics Forum 36*, 8 (2017), 354–368. 2, 3, 4, 10

[KAGS19] KIM B., AZEVEDO V. C., GROSS M., SOLENTHALER B.: Transport-based neural style transfer for smoke simulations. *ACM Transactions on Graphics 38*, 6 (nov 2019), 1–11. URL: http://dl.acm.org/citation.cfm?doid=3355089.3356560, doi:10.1145/3355089.3356560. 3

[KAT*19] KIM B., AZEVEDO V. C., THUEREY N., KIM T., GROSS M., SOLENTHALER B.: Deep Fluids: A Generative Network for Parameterized Fluid Simulations. *Computer Graphics Forum 38*, 2 (2019). 9

[KB14] KINGMA D. P., BA J.: Adam: A Method for Stochastic Optimization. URL: http://arxiv.org/abs/1412.6980, arXiv:1412.6980. 2

[KMT06] KIM Y., MACHIRAJU R., THOMPSON D.: Path-Based Control of Smoke Simulations. In *Proceedings of the 2006 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Goslar, DEU, 2006), SCA '06, Eurographics Association, pp. 33–42. 3

[KSH17] KRIZHEVSKY A., SUTSKEVER I., HINTON G. E.: ImageNet classification with deep convolutional neural networks. *Communications of the ACM 60*, 6 (may 2017), 84–90. URL: http://dl.acm.org/citation.cfm?doid=3098997.3065386, doi:10.1145/3065386. 2

[KTJG08] KIM T., THÜREY N., JAMES D., GROSS M.: Wavelet turbulence for fluid simulation. In *ACM Transactions on Graphics (TOG)* (2008), vol. 27, ACM, p. 50. 6

[LADL18] LI T.-M., AITTALA M., DURAND F., LEHTINEN J.: Differentiable Monte Carlo ray tracing through edge sampling. In *SIGGRAPH Asia 2018 Technical Papers on - SIGGRAPH Asia '18* (New York, New York, USA, 2018), ACM Press, pp. 1–11. URL: http://dl.acm.org/citation.cfm?doid=3272127.3275109, doi:10.1145/3272127.3275109. 2

[LBK17] LIU T., BOUAZIZ S., KAVAN L.: Quasi-Newton Methods for Real-Time Simulation of Hyperelastic Materials. *ACM Transactions on Graphics 36*, 4 (may 2017), 1. URL: http://dl.acm.org/citation.cfm?doid=3072959.2990496, doi:10.1145/3072959.2990496. 2

[LLK19] LIANG J., LIN M., KOLTUN V.: Differentiable cloth simulation for inverse problems. In *Advances in Neural Information Processing Systems 32*, Wallach H., Larochelle H., Beygelzimer A., d'Alché-Buc F., Fox E., Garnett R., (Eds.). Curran Associates, Inc., 2019, pp. 771–780. 3

[MCP*09] MULLEN P., CRANE K., PAVLOV D., TONG Y., DESBRUN M.: Energy-preserving integrators for fluid animation. *ACM Transactions on Graphics 28*, 3 (jul 2009), 1. URL: http://portal.acm.org/citation.cfm?doid=1531326.1531344, doi:10.1145/1531326.1531344. 3

[MM13] MADILL J., MOULD D.: Target Particle Control of Smoke Simulation. In *Proceedings of Graphics Interface 2013* (CAN, 2013), GI '13, Canadian Information Processing Society, pp. 125–132. 3

[MMS04] MIHALEF V., METAXAS D., SUSSMAN M.: Animation and control of breaking waves. In *Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation - SCA '04* (New York, New York, USA, 2004), ACM Press, p. 315. URL: http://portal.acm.org/citation.cfm?doid=1028523.1028565, doi:10.1145/1028523.1028565. 3

[MTGG11] MARTIN S., THOMASZEWSKI B., GRINSPUN E., GROSS M.: Example-based elastic materials. *ACM Transactions on Graphics 30*, 4 (jul 2011), 1. URL: http://portal.acm.org/citation.cfm?doid=2010324.1964967, doi:10.1145/2010324.1964967. 2

[MTPS04] MCNAMARA A., TREUILLE A., POPOVIĆ Z., STAM J.: Fluid control using the adjoint method. In *ACM SIGGRAPH 2004 Papers on - SIGGRAPH '04* (2004), ACM Press, p. 449. 2, 3, 4, 6

[MVW*16] MANTEAUX P.-L., VIMONT U., WOJTAN C., ROHMER D., CANI M.-P.: Space-time sculpting of liquid animation. In *Proceedings of the 9th International Conference on Motion in Games - MIG '16* (New York, New York, USA, 2016), ACM Press, pp. 61–71. URL: http://dl.acm.org/citation.cfm?doid=2994258.2994261, doi:10.1145/2994258.2994261. 3

[NB11] NIELSEN M. B., BRIDSON R.: Guide shapes for high resolution naturalistic liquid simulation. In *ACM SIGGRAPH 2011 papers on - SIGGRAPH '11* (New York, New York, USA, 2011), ACM Press, p. 1. URL: http://portal.acm.org/citation.cfm?doid=1964921.1964978, doi:10.1145/1964921.1964978. 3

[NC10] NIELSEN M. B., CHRISTENSEN B. B.: Improved Variational Guiding of Smoke Animations. *Computer Graphics Forum 29*, 2 (may 2010), 705–712. URL: http://doi.wiley.com/10.1111/j.1467-8659.2009.01640.x, doi:10.1111/j.1467-8659.2009.01640.x. 3

[NCZ*09] NIELSEN M. B., CHRISTENSEN B. B., ZAFAR N. B., ROBLE D., MUSETH K.: Guiding of smoke animations through variational coupling of simulations at different resolutions. In *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation - SCA '09* (New York, New York, USA, 2009), ACM Press, p. 217. URL: http://portal.acm.org/citation.cfm?doid=1599470.1599499, doi:10.1145/1599470.1599499. 2, 3

[Nes83] NESTEROV Y. E.: A method for solving the convex programming problem with convergence rate O(1/k^2). *Dokl. Akad. Nauk SSSR 269* (1983), 543–547. URL: http://ci.nii.ac.jp/naid/10029946121/en/. 2

[NSO12] NARAIN R., SAMII A., O'BRIEN J. F.: Adaptive anisotropic remeshing for cloth simulation. *ACM Transactions on Graphics 31*, 6 (nov 2012), 1. URL: http://dl.acm.org/citation.cfm?doid=2366145.2366171, doi:10.1145/2366145.2366171. 2

[NVT*14] NEUMANN T., VARANASI K., THEOBALT C., MAGNOR M., WACKER M.: Compressed Manifold Modes for Mesh Processing. *Computer Graphics Forum 33*, 5 (aug 2014), 35–44. URL: http://doi.wiley.com/10.1111/cgf.12429, doi:10.1111/cgf.12429. 2

[OBLN17] OVERBY M., BROWN G. E., LI J., NARAIN R.: ADMM Projective Dynamics: Fast Simulation of Hyperelastic Models with Dynamic Constraints. *IEEE Transactions on Visualization and Computer Graphics 23*, 10 (oct 2017), 2222–2234. URL: http://ieeexplore.ieee.org/document/7990052/, doi:10.1109/TVCG.2017.2730875. 2

[PCS04] PIGHIN F., COHEN J. M., SHAH M.: Modeling and editing flows using advected radial basis functions. In *Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation - SCA '04* (New York, New York, USA, 2004), ACM Press, p. 223. URL: http://portal.acm.org/citation.cfm?doid=1028523.1028552, doi:10.1145/1028523.1028552. 3

[PGM*19] PASZKE A., GROSS S., MASSA F., LERER A., BRADBURY J., CHANAN G., KILLEEN T., LIN Z., GIMELSHEIN N., ANTIGA L., DESMAISON A., KOPF A., YANG E., DEVITO Z., RAISON M., TEJANI A., CHILAMKURTHY S., STEINER B., FANG L., BAI J., CHINTALA S.: Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, Wallach H., Larochelle H., Beygelzimer A., d'Alché-Buc F., Fox E., Garnett R., (Eds.). Curran Associates, Inc., 2019, pp. 8024–8035. 6

[PHT*13] PAN Z., HUANG J., TONG Y., ZHENG C., BAO H.: Interactive localized liquid motion editing. *ACM Transactions on Graphics 32*, 6 (nov 2013), 1–10. URL: http://dl.acm.org/citation.cfm?doid=2508363.2508429, doi:10.1145/2508363.2508429. 3

[PM17a] PAN Z., MANOCHA D.: Editing smoke animation using a deforming grid. *Computational Visual Media 3*, 4 (dec 2017), 369–378. URL: http://link.springer.com/10.1007/s41095-017-0096-2, doi:10.1007/s41095-017-0096-2. 3

[PM17b] PAN Z., MANOCHA D.: Efficient solver for spacetime control

of smoke. *ACM Transactions on Graphics (TOG) 36*, 4 (2017), 1. 2, 3, 4, 6, 7, 8

[PSE*00] POPOVIĆ J., SEITZ S. M., ERDMANN M., POPOVIĆ Z., WITKIN A.: Interactive manipulation of rigid body simulations. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques - SIGGRAPH '00* (2000), pp. 209–217. 3

[REN*04] RASMUSSEN N., ENRIGHT D., NGUYEN D., MARINO S., SUMNER N., GEIGER W., HOON S., FEDKIW R.: Directable photorealistic liquids. In *Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation - SCA '04* (New York, New York, USA, 2004), ACM Press, p. 193. URL: http://portal.acm.org/citation.cfm?doid=1028523.1028549, doi:10.1145/1028523.1028549. 3

[Ric22] RICHARDSON L. F.: *Weather prediction by numerical process*. Cambridge university press, 1922. 5

[RLL*13] REN B., LI C.-F., LIN M. C., KIM T., HU S.-M.: Flow Field Modulation. *IEEE Transactions on Visualization and Computer Graphics 19*, 10 (oct 2013), 1708–1719. URL: http://ieeexplore.ieee.org/document/6502161/, doi:10.1109/TVCG.2013.73. 3

[RTWT12] RAVEENDRAN K., THUEREY N., WOJTAN C., TURK G.: Controlling Liquids Using Meshes. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Goslar Germany, Germany, 2012), SCA '12, Eurographics Association, pp. 255–264. URL: http://dl.acm.org/citation.cfm?id=2422356.2422393. 2, 3

[RWTT14] RAVEENDRAN K., WOJTAN C., THUEREY N., TURK G.: Blending liquids. *ACM Transactions on Graphics 33*, 4 (jul 2014), 1–10. URL: http://dl.acm.org/citation.cfm?doid=2601097.2601126, doi:10.1145/2601097.2601126. 3

[SA07] SORKINE O., ALEXA M.: As-rigid-as-possible Surface Modeling. In *Proceedings of the Fifth Eurographics Symposium on Geometry Processing* (Aire-la-Ville, Switzerland, Switzerland, 2007), SGP '07, Eurographics Association, pp. 109–116. URL: http://dl.acm.org/citation.cfm?id=1281991.1282006. 2

[SCOL*04] SORKINE O., COHEN-OR D., LIPMAN Y., ALEXA M., RÖSSL C., SEIDEL H.-P.: Laplacian surface editing. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing - SGP '04* (New York, New York, USA, 2004), ACM Press, p. 175. URL: http://portal.acm.org/citation.cfm?doid=1057432.1057456, doi:10.1145/1057432.1057456. 2

[SDE05] SCHPOK J., DWYER W., EBERT D. S.: Modeling and Animating Gases with Simulation Features. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (New York, NY, USA, 2005), SCA '05, Association for Computing Machinery, pp. 97–105. URL: https://doi.org/10.1145/1073368.1073381, doi:10.1145/1073368.1073381. 3

[SDKN18] SATO S., DOBASHI Y., KIM T., NISHITA T.: Example-based turbulence style transfer. *ACM Transactions on Graphics (TOG) 37*, 4 (2018), 84. 3

[SDN18] SATO S., DOBASHI Y., NISHITA T.: Editing Fluid Animation Using Flow Interpolation. *ACM Transactions on Graphics 37*, 5 (sep 2018), 1–12. URL: http://dl.acm.org/citation.cfm?doid=3278329.3213771, doi:10.1145/3213771. 3

[SF93] STAM J., FIUME E.: Turbulent wind fields for gaseous phenomena. In *Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques* (New York, NY, USA, 1993), SIGGRAPH '93, Association for Computing Machinery, p. 369–376. URL: https://doi.org/10.1145/166117.166163, doi:10.1145/166117.166163. 6

[SF18] SCHENCK C., FOX D.: Spnets: Differentiable fluid dynamics for deep neural networks. In *2nd Annual Conference on Robot Learning, CoRL 2018* (2018), vol. 87 of *Proceedings of Machine Learning Research*, PMLR. 3

[SKF11] SIDERIS C., KAPADIA M., FALOUTSOS P.: Parallelized incomplete poisson preconditioner in cloth simulation. In *International Conference on Motion in Games* (2011), Springer, pp. 389–399. 6

[SY05a] SHI L., YU Y.: Controllable smoke animation with guiding objects. *ACM Transactions on Graphics 24*, 1 (jan 2005), 140–164. URL: http://portal.acm.org/citation.cfm?doid=1037957.1037965, doi:10.1145/1037957.1037965. 3

[SY05b] SHI L., YU Y.: Taming liquids for rapidly changing targets. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation - SCA '05* (2005), ACM Press, p. 229. URL: http://portal.acm.org/citation.cfm?doid=1073368.1073401, doi:10.1145/1073368.1073401. 3

[Thu16] THUEREY N.: Interpolations of smoke and liquid simulations. *ACM Transactions on Graphics (TOG) 36*, 1 (2016), 1–16. 3

[TKPR09] THÜREY N., KEISER R., PAULY M., RÜDE U.: Detail-preserving fluid control. *Graphical Models 71*, 6 (nov 2009), 221–228. URL: https://linkinghub.elsevier.com/retrieve/pii/S1524070309000022, doi:10.1016/j.gmod.2008.12.007. 3

[TL19] TAKAHASHI T., LIN M. C.: Video-guided real-to-virtual parameter transfer for viscous fluids. *ACM Transactions on Graphics 38*, 6 (nov 2019), 1–12. URL: http://dl.acm.org/citation.cfm?doid=3355089.3356551, doi:10.1145/3355089.3356551. 3

[TMPS03] TREUILLE A., MCNAMARA A., POPOVIĆ Z., STAM J.: Keyframe control of smoke simulations. *ACM Transactions on Graphics 22*, 3 (jul 2003), 716. URL: http://portal.acm.org/citation.cfm?doid=882262.882337, doi:10.1145/882262.882337. 2, 3, 4

[TNGF15] TOURNIER M., NESME M., GILLES B., FAURE F.: Stable constrained dynamics. *ACM Transactions on Graphics 34*, 4 (jul 2015), 132:1–132:10. URL: http://dl.acm.org/citation.cfm?doid=2809654.2766969, doi:10.1145/2766969. 2

[WP10] WEISSMANN S., PINKALL U.: Filament-based smoke with vortex shedding and variational reconnection. *ACM Transactions on Graphics 29*, 4 (jul 2010), 1. URL: http://portal.acm.org/citation.cfm?doid=1778765.1778852, doi:10.1145/1778765.1778852. 3

[YCZ11] YUAN Z., CHEN F., ZHAO Y.: Pattern-guided smoke animation with lagrangian coherent structure. *ACM Transactions on Graphics 30*, 6 (dec 2011), 1. URL: http://dl.acm.org/citation.cfm?doid=2070781.2024170, doi:10.1145/2070781.2024170. 3

[YLYJ13] YANG B., LIU Y., YOU L., JIN X.: A unified smoke control method based on signed distance field. *Computers & Graphics 37*, 7 (nov 2013), 775–786. URL: https://linkinghub.elsevier.com/retrieve/pii/S0097849313000757, doi:10.1016/j.cag.2013.05.001. 3

[ZBLN97] ZHU C., BYRD R. H., LU P., NOCEDAL J.: Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization. *ACM Transactions on Mathematical Software 23*, 4 (dec 1997), 550–560. URL: http://portal.acm.org/citation.cfm?doid=279232.279236, doi:10.1145/279232.279236. 2

[ZPOD19] ZHANG J., PENG Y., OUYANG W., DENG B.: Accelerating ADMM for efficient simulation and optimization. *ACM Transactions on Graphics 38*, 6 (nov 2019), 1–21. URL: http://dl.acm.org/citation.cfm?doid=3355089.3356491, doi:10.1145/3355089.3356491. 2

[ZZ18] ZHANG Y., ZAVLANOS M. M.: A Consensus-Based Distributed Augmented Lagrangian Method. In *2018 IEEE Conference on Decision and Control (CDC)* (dec 2018), IEEE, pp. 1763–1768. URL: https://ieeexplore.ieee.org/document/8619512/, doi:10.1109/CDC.2018.8619512. 2