Model-Reduced Variational Fluid Simulation

Beibei Liu MSU
Gemma Mason Caltech/U. Auckland
Julian Hodgson UCL
Yiying Tong MSU
Mathieu Desbrun Caltech-INRIA

Abstract

We present a model-reduced variational Eulerian integrator for incompressible fluids, which combines the efficiency gains of dimension reduction, the qualitative robustness of coarse spatial and temporal resolutions of geometric integrators, and the simplicity of sub-grid accurate boundary conditions on regular grids to deal with arbitrarily-shaped domains. At the core of our contributions is a functional map approach to fluid simulation for which scalar- and vector-valued eigenfunctions of the Laplacian operator can be easily used as reduced bases. Using a variational integrator in time to preserve liveliness and a simple, yet accurate embedding of the fluid domain onto a Cartesian grid, our model-reduced fluid simulator can achieve realistic animations in significantly less computational time than full-scale non-dissipative methods but without the numerical viscosity from which current reduced methods suffer. We also demonstrate the versatility of our approach by showing how it easily extends to magnetohydrodynamics and turbulence modeling in 2D, 3D and curved domains.

CR Categories: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation.

Keywords: Computational fluid dynamics, model reduction, Eulerian simulation, energy preservation, sub-grid-resolution geometry.

1 Introduction

Accurate simulation of incompressible fluids is a well-studied topic in computational fluid dynamics. Fluid animation research is driven by a different emphasis: in the context of computer graphics, the focus is on capturing the visual complexity of typical incompressible fluid motions (such as vortices and volutes) with minimum computational cost. However, this relentless quest for efficiency has often resulted in time integrators that exhibit large numerical viscosity [Stam 1999], as they proceed via operator splitting through advection followed by divergence-free projection. The incurred numerical dissipation has also, besides its obvious visual artifacts, the unintended consequence that previews on coarse spatial and temporal resolutions are far from predictive of the final, high-resolution run. Non-dissipative methods have been proposed more recently [Mullen et al. 2009]; however, they require large, non-linear solves, hampering efficiency. On the other hand, model-reduced integrators [Treuille et al. 2006] manipulate a smaller set of degrees of freedom found via Galerkin dimension reduction to capture the main components of the flow efficiently, at the cost of excessive vorticity smearing. Similarly, regular spatial grids are often preferred due to their significantly lighter data structures and sparser stencils—yet, their use conflicts with the proper treatment of boundary conditions over complex, non-grid-aligned domains. While pressure projection with subgrid accuracy have been recently proposed on Cartesian grids [Batty et al. 2007; Ng et al. 2009], non-dissipative methods still require boundary-conforming meshes.

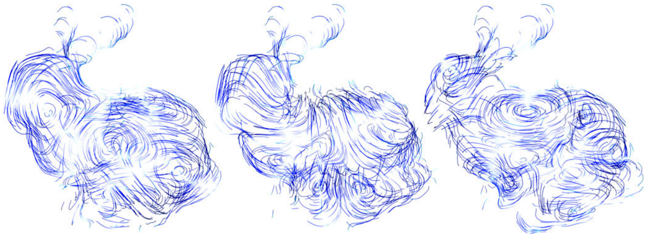


Figure 1: ***Model-reduced fluids on regular grids.*** *Our energy-preserving approach integrates a fluid flow variationally using a small number of divergence-free velocity field bases over an arbitrary domain (visualized here are the* 5^{th}*,* 10^{th}*, and* 15^{th} *eigenvectors of the* 2*-form Laplacian) computed with subgrid accuracy on a regular grid (here, a* $42\times42\times32$ *grid). Our integrator is versatile: it can be used for realtime fluid animation, magnetohydrodynamics, and turbulence models, with either explicit or implicit integration.*

In this paper, we introduce a variational model-reduced Eulerian fluid solver with sub-grid accuracy which bypasses the traditional numerical curses of Galerkin projected dynamics, while keeping the efficiency of Cartesian grid-based simulation.

1.1 Background

Early computer animation Eulerian methods for incompressible fluid simulation were based on explicit finite differences [Foster and Metaxas 1997] which suffered from the slow convergence of their iterative approach to divergence-free projection. Stam [1999] introduced semi-Lagrangian advection and a sparse Poisson solver which brought much improved efficiency and stability. However, these improvements came at the cost of significant dissipation—a common issue that one can partially mitigate via vorticity confinement [Steinhoff and Underhill 1994], reinjection of vorticity with particles [Selle et al. 2005], or curl correction [Zhang et al. 2015]. Significantly less dissipative time integrators were also proposed through semi-Lagrangian advection of vorticity [Elcott et al. 2007], or even energy-preserving methods [Mullen et al. 2009]. However, these improved numerical methods often carry higher computational costs. Consequently, coupled Eulerian-Lagrangian (hybrid) methods (see, for instance, [Losasso et al. 2008; Golas et al. 2012]) have flourished recently, as they offer a good compromise between efficiency and dissipation.

Handling boundaries well is also crucial for incompressible fluids, as boundary layers can significantly impact fluid motion. Avoiding the staircase effects that voxelized domains generate was achieved using simplicial meshes or hybrid meshes [Feldman et al. 2005], but irregular connectivity often affects the efficiency of the solvers involved. Inspired by the immersed boundary and interface methods, the use of regular grids with modified numerical operators to handle arbitrary domains was proposed in [Batty et al. 2007], then made convergent by [Ng et al. 2009] while maintaining symmetry of the solves needed by the integrators. Another approach using virtual nodes was also proposed recently [Howes et al. 2013].

Fluid simulation over non-flat domains has received significant attention as well. Most notably, Stam adapted his Stable Fluid method to handle curvilinear coordinates [Stam 2003], while Azencot et al. [2014] recently proposed to use the Lie derivative operator representation in the spectral domain to represent a velocity field on an arbitrary surface, and performed advection of vorticity through a linearized exponential map of the operator representation. Methods that are using only intrinsic operators can also handle curved domains without alterations [Elcott et al. 2007; Mullen et al. 2009].

While most Eulerian methods use a finite-dimensional description of the fluid using DOFs on cell faces or centers, model reduction was also introduced in an effort to approximate the fluid motion using only a small number of basis functions. The early days of computational fluid dynamics for atmospheric simulation proposed to reduce complexity by discarding high frequencies through the use of a low number of modes (typically, harmonics or spherical harmonics) to describe the vector field [Silberman 1954; Yudovich 1963], while pseudo-spectral methods leveraged fast conversion between modal coefficients and spatial representation via the Fast Fourier Transform for highly symmetric domains [Orszag 1969]. Dimensionality reduction was first introduced for fluid animation by Treuille et al. [2006] through Galerkin projection onto a reduced set of basis functions computed through principal component analysis of a training set of fluid motions. Their method was demonstrated on regular grids, but is generalizable to tetrahedral meshes. A number of works followed, proposing the use of different bases such as Legendre polynomials [Gupta and Narasimhan 2007], trigonometric functions [Long and Reinhard 2009], or even non-polynomial Galerkin projection [Stanton et al. 2013]; eventually, Laplacian eigenvectors were pointed out by [De Witt et al. 2012] to be particularly appropriate harmonics as they guarantee divergence free flows and facilitate the conversion between vorticity and velocity, while offering a sparse advection operator for symmetric domains. These eigenfunctions also allow easy implementation of viscosity, and eliminate the need for training sets of velocity fields. For simulations involving moving solids, model reduction can also be conducted on a moving grid [Cohen et al. 2010]. The use of cubature, initially proposed to achieve model-reduced simulation of elastic models [An et al. 2008; von Tycowicz et al. 2013; Li et al. 2014], can speed up re-simulation of fluids in a reduced subspace as well [Kim and Delaney 2013]. However, the gain in efficiency of all such model-reduced simulations is often counterbalanced by (at times severe) energy or vorticity dissipation and the need for unstructured meshes to capture complex boundaries.

1.2 Contributions

In this paper, we formulate a model-reduced variational fluid integrator that combines the benefits of *non-dissipative* integrators with the use of *dimension reduction* and *Cartesian grids* over arbitrary domains. Based on a description of the fluid motion through functional maps, a variational integrator is derived from Hamilton's principle [Marsden and West 2001; Kharevych et al. 2006], resulting in a Lie algebra integrator with non-holonomic constraints [Pavlov et al. 2011; Gawlik et al. 2011]. We use spectral approximation of the functional map through (cell-based) scalar and (face-based) vector Laplacian eigenvectors in order to offer model reduction without losing the variational properties of the integrator, with controllable energy cascading. This setup allows us to use not only low frequencies to capture the basic behavior of a flow, but also a few selected higher frequencies to add realism at low cost. Furthermore, we extend the embedded-boundary approach of [Ng et al. 2009] to our framework in order to compute spectral (scalar- and vector-valued) basis functions of arbitrary domains directly on *regular grids* for fast computations with sub-grid accuracy. Finally, our approach uses the typical Eulerian setup of flux-based solvers; consequently, addition of fine details through spectral noise [Stam and Fiume 1993], wavelet [Kim et al. 2008], empirical mode decomposition [Gao et al. 2013], subgrid turbulence [Schechter and Bridson 2008; Narain et al. 2008], curl correction [Zhang et al. 2015], or through enforcing Lagrangian coherent structure [Yuan et al. 2011] can be done straightforwardly. We demonstrate the efficiency of our resulting integrator through a number of examples in 2D, 3D, and curved 2D domains, as well as its versatility by pointing out how to extend its use to magnetohydrodynamics, subgrid scale models, and other fluid equations. Our paper thus extends the variational approach of [Mullen et al. 2009; Gawlik et al. 2011] to arbitrary reduced bases, adopts the (now Eulerian) vorticity advection of [Elcott and Schröder 2006], and offers a structure-preserving version of the Laplacian-based integrator of [De Witt et al. 2012].

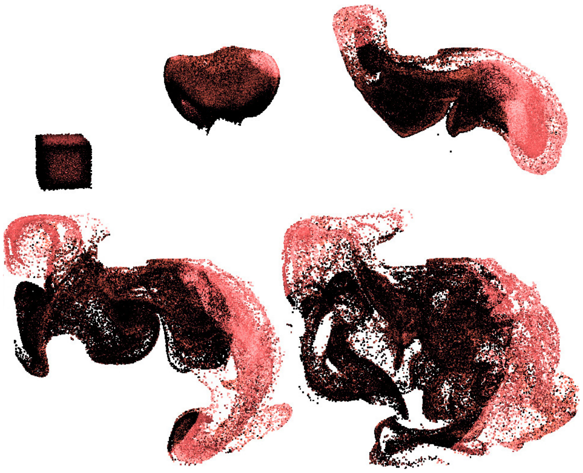


Figure 2: ***3D bunny buoyancy test****: A hot cube of air initially located at the center of a 3D bunny-shaped domain is advected through buoyancy. Computations were performed using a modified Hodge star on a* $42\times42\times32$ *grid, with only* 100 *modes.*

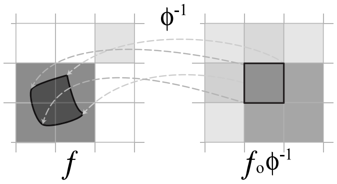
Note that our contributions are numerical in nature. They do not target improvements in visual complexity, but in efficiency (through embedding of arbitrary boundaries on Cartesian grids, §3.4), generality (arbitrary reduced bases can be employed, §3.6), and controllability (energy cascading and viscosity are consistent across temporal and spatial scales, §3.5).

2 Recap of Variational Eulerian Integration

In order to provide fluid simulations with stable long-term behavior across different space or time resolutions, Pavlov et al. [2011] introduced a variational integrator for fluids in Eulerian representation by discretizing the fluid motion as a Lie group acting on the space of functions, and formulating the kinetic energy on its Lie algebra. The motion of an incompressible, inviscid fluid is described in the continuous setting by a volume-preserving flow ϕ_t, i.e., a particle which is at a point p at time $t=0$ will be found at $\phi_t(p)$ after being advected by the flow. The set of all such possible flows is given by the set of volume-preserving maps ϕ_t from the domain to itself. This set having the structure of an infinite-dimensional Lie group, it was discretized into a finite-dimensional Lie group for computational purposes. Moving from a Lie group to the associated Lie algebra connects the Lie group viewpoint of flows and "functional maps" [Azencot et al. 2014] to the Lie algebra viewpoint of vector fields, as we now briefly review.

2.1 Discretization process

We assume that the fluid domain is discretized as a mesh. Without loss of generality, we restrict our discussion to regular grids for simplicity, as we will show in Sec 3.4 how to embed arbitrary domains into a Cartesian grid. We discretize a continuous function $f(\mathbf{x})$ on our space by taking an average (integrated) value f_i per grid cell i of the mesh, which we arrange in a vector $\mathbf{f}$. This definition of discrete functions allows us to discretize the set of possible flows ϕ_t using a *functional map* (or Koopman operator) $(f \circ \phi_t^{-1})(\mathbf{x}) = f(\phi_t^{-1}(\mathbf{x}))$. Specifically, this functional map is encoded as a matrix q of size the square of the number of cells, representing the action of ϕ_t on any discrete function; that is, the integrated values $\mathbf{f}$ of f per cell become $q\mathbf{f}$ once f is advected by the flow ϕ. Because the discrete flow acts as a functional map, it should always take the constant function to itself. That is, for all q, we require that $q\mathbf{1} = \mathbf{1}$, where $\mathbf{1}$ is a vector of ones (see [Pavlov et al. 2011] for the equivalent condition on an arbitrary mesh). This is the same as saying that the row sums of q are equal to 1, i.e., q is *signed stochastic*. Since we are simulating an incompressible fluid, we also require that the discrete flow be volume-preserving. This condition is achieved by asking that the discrete flow preserves the inner product of vectors, that is, q is *orthogonal*, i.e., $q^t = q^{-1}$. Thus, we find that we need to take q to be an element of the Lie group G of orthogonal, signed stochastic matrices. This matrix group represents our discrete fluid configuration, as we describe next.



2.2 The Eulerian Lie Algebra viewpoint

We can view the finite-dimensional Lie group G as a configuration space: it encodes the space of possible "positions" for the discrete fluid, in that each element of the Lie group represents a possible way that the fluid could have evolved from its initial position. This Lie group represents a Lagrangian perspective as it identifies the fluid particles in a given cell by recording which cells they originally came from. The associated Eulerian perspective is given by the Lie algebra $\mathfrak{g}$ of matrices of the form $\dot{q} \circ q^{-1}$ for $q \in G$. It was shown in [Pavlov et al. 2011] that any matrix $A \in \mathfrak{g}$ of this Lie algebra is both *antisymmetric* ($A^t = -A$) and *row-null* ($A\mathbf{1} = \mathbf{0}$), and corresponds to a discrete counterpart of the Lie derivative L_v with respect to the continuous velocity field $v = \dot{\phi} \circ \phi^{-1}$. Thus, the product $A\mathbf{f}$ of such a matrix with a discrete function $\mathbf{f}$ approximates the continuous term $v \cdot \nabla f$. Furthermore, if cells i and j are nearest neighbors, then the matrix element A_{ij} represents the flux of the fluid through the face shared by cells i and j. Thus, an element of the Lie algebra $\mathfrak{g}$ of G is directly linked to the usual flux-based (Marker And Cell) discretization of vector field in fluid simulators [Harlow and Welch 1965].

2.3 Non-holonomic constraint

Whilst the elements A_{ij} for $A \in \mathfrak{g}$ have a clear physical interpretation in the case where i and j are nearest neighbors, this is not the case for elements representing interactions between cells that are not immediate neighbors. Similarly to a CFL condition, we prohibit fluid particles from skipping to non-neighboring cells, by restricting the Lie algebra to the constrained set $\mathcal{S}$, the set of matrices A such that $A_{ij} = 0$ unless cells i and j share a face (or an edge in 2D). We require the elements of $\mathfrak{g}$ that we use to represent the fluid velocity fields to fall into this constrained set. This has the additional advantage of making the matrices sparse, dramatically decreasing the amount of memory required and the computational time, as much fewer degrees of freedom need to be updated—and now, the traditional MAC discretization with fluxes corresponds exactly to a Lie algebra element in this constrained set.

Constraining the matrices in this way requires a non-holonomic constraint, because the set $\mathcal{S}$ is not closed under the Lie bracket. That is, interactions between nearest neighbors followed by further interactions between nearest neighbors produce interactions between cells that are two-away from each other, which are therefore not inside the constrained set $\mathcal{S}$.

2.4 Creating a variational numerical method

Using this discretization, one can create a variational numerical method for ideal, incompressible fluids through the Euler-Poincaré equations [Gawlik et al. 2011] for the Lagrangian given by

$$\mathcal{L}_{\text{Euler}} = \frac{1}{2}\langle A, A\rangle \approx \frac{1}{2}\int v^2 \,\mathrm{d}x, \tag{1}$$

and subject to the non-holonomic constraint $A \in \mathcal{S}$. The resulting numerical method exhibits no numerical dissipation, and produces good qualitative behavior over long timescales. Changing the time integration scheme to be time reversible leads to exact energy preservation [Mullen et al. 2009]. With control over dissipation and robustness to time step and grid size, this computational tool greatly facilitates the design of fluid animation. Note that this variational integrator also guarantees that the relabeling symmetry implies a discrete version of Kelvin's circulation theorem, i.e., circulation of velocity field (represented as a Lie algebra element) along a closed loop (represented as a 1-cycle [Pavlov et al. 2011]) transported along the fluid flow is invariant, which helps keep the vivid details of vorticity in the fluid simulation without resorting to additional energy-injecting measures such as vorticity confinement, as shown in [Elcott et al. 2007; Mullen et al. 2009]. However, the time integration requires a quadratic solve based on *all* the fluxes in the domain, making it inappropriate for realtime simulation.

3 Model-reduced Variational Integrator

We present an integrator which extends the approach of [Pavlov et al. 2011], using a different functional-map Lie group, similarly interlinked with an Eulerian velocity-based Lie algebra picture. Our method offers the additional advantage of fast computations on arbitrary domains: we use *reduced coordinates* to encode the most significant components of the spatial scalar and vector fields, and perform subgrid accurate precomputations on simple *regular grids*. We will focus on Euler equations first, before discussing variants such as Navier-Stokes and magnetohydrodynamics (MHD).

3.1 Spectral Bases

We first define the discrete, *reduced* scalar and velocity fields on which our functional map Lie group will act. Extending what was advocated in [De Witt et al. 2012], we use the orthonormal bases for 2-forms and 3-forms given by the eigenfunctions of the deRham-Laplacian operators on an arbitrary discrete mesh $\mathcal{M}$. These are calculated using the discrete operators of Discrete Exterior Calculus [Desbrun et al. 2008; Arnold et al. 2006], allowing us to leverage the large literature on their implementation and structure-preserving properties. From this small set of basis functions, we efficiently encode through reduced coordinates the full-space fields typically used in the MAC scheme, i.e., fluxes through cell boundaries (discrete 2-forms) to represent velocity fields, and densities integrated in each cell (discrete 3-forms) to represent scalar fields (such as smoke density or geostrophic momentum in rotating stratified flow [Desbrun et al. 2013]).

Choice of bases. We denote the i-th eigenfunction of the 3-form Laplacian Δ_3 as Φ_i, with associated eigenvalue $-\mu_i^2$,

$$\Delta_3 \Phi_i = -\mu_i^2 \Phi_i.$$

The eigenfunctions corresponding to the $M_3{+}1$ *smallest* μ_i can be assembled into a low-frequency basis

$$\{\Phi_0, ..., \Phi_{M_3}\}.$$

Note that depending on the boundary condition, $\mu_0 = 0$ may correspond to more than one harmonic function; but these remain stationary when advected by divergence-free velocity fields with zero flux across the boundary, and are thus omitted in our discussion.

Similarly, we denote the i-th eigenfunction of the 2-form Laplacian Δ_2 as Ψ_i, with its associated eigenvalue $-\kappa_i^2$:

$$\Delta_2 \Psi_i = -\kappa_i^2 \Psi_i.$$

We also assemble the first M_2 eigenvector fields (corresponding to the M_2 smallest κ_i) into a finite dimensional low-frequency basis,

$$\{\Psi_1, \ldots \Psi_{M_2}\}.$$

Some of the 2-form eigenfunctions are not divergence-free, and these eigenfunctions can be identified as gradient fields, $\nabla\Phi_i/\mu_i$ (see §A). Thus, we can reorder the eigenfunctions of Δ_2 into

$$\{h_1, \ldots, h_{\beta_1}, \frac{\nabla\Phi_1}{\mu_1}, \ldots, \frac{\nabla\Phi_{M_3}}{\mu_{M_3}}, \Psi_1, \ldots \Psi_{M_C}\},$$

where h_i are harmonic 2-forms (corresponding to frequency $\kappa_i = 0$) with β_1 being the first Betti number determined by the topology of the domain (basically, the number of tunnels plus the number of connected components of the boundary minus one), and $M_C = M_2 - M_3 - \beta_1$ denoting the number of non-harmonic but divergence-free basis functions.

Discretization. Computing our spectral bases requires a proper discretization of the Laplacian operators and of boundary conditions. Both topics are well studied, and many implementations can be leveraged [Elcott and Schröder 2006; Bell and Hirani 2008]. In particular, we note that discrete Laplacians are typically *integrated* Laplacians, meaning that the two eigenvalue problems mentioned above are discretized as two generalized eigenvalue problems

$$(\star_3 \Delta_3)\Phi_i = -\mu_i^2 \star_3 \Phi_i \quad \text{and} \quad (\star_2 \Delta_2)\Psi_i = -\kappa_i^2 \star_2 \Psi_i$$

respectively, to make the discrete operators symmetric and thus allow for efficient numerical solvers. We provide a detailed guide to discretization on *arbitrary unstructured meshes* in §A to explicate how to enforce no-transfer and free-slip conditions (corresponding, respectively, to $v_n|_{\partial\mathcal{M}} = 0$ and $\partial v_t/\partial n\,|_{\partial\mathcal{M}} = 0$ if the continuous velocity field is decomposed at the boundary into its normal and tangential components, $v = v_n + v_t$). Note that only two operators are required: the exterior derivative d and the Hodge star $\star$. The first operator is purely topological, while the second is just a scaling operation per edge, face, and cell. Moreover, we will see in §3.4 that this latter operator can be trivially modified to handle arbitrary fluid domains without having to use anything else but a regular grid. From these two operators, both Laplacians are easily assembled, and low-frequency eigenfields are found via Lanczos iterations.

3.2 Spectral Lie group

While earlier methods [Pavlov et al. 2011; Gawlik et al. 2011] have defined scalar fields using a spatial representation through linear combinations of locally-supported piecewise-constant basis functions, we use a spectral representation through linear combinations of the aforementioned spectral basis functions Φ_i, allowing us to drastically reduce the number of degrees of freedom the integrator will have to update, while still conforming to the shape of the domain (see Fig. 3).

Lie group. We encode the fluid motion through a time-varying Lie group element $q(t)$ that represents a functional map induced by the fluid flow ϕ_t, mapping a function $f(\mathbf{x}) = \sum_i f_i \Phi_i(\mathbf{x})$ linearly to another function $g(\mathbf{x}) = \sum_i g_i \Phi_i(\mathbf{x})$ such that $g(\mathbf{x}) = f \circ \phi^{-1}(\mathbf{x})$. As the function space is approximated by a finite dimensional space spanned by low-frequency basis functions, q can be encoded by a $(M_3+1)\times(M_3+1)$ matrix. The volume-preserving property of the flow still implies the orthogonality of the matrix, i.e, $q^t q = \text{Id}$. So we are looking for a subgroup of $O(M_3+1)$, or, more accurately, of $SO(M_3+1)$, since we wish to describe gradual changes from the identity. The condition that constant functions are mapped to themselves in this low-frequency Lie group becomes $q_{0i} = \delta_{0i}$ and $q_{i0} = \delta_{i0}$, where δ_{ij} is the Kronecker symbol, since 0-th frequency represents the constant function. This effectively removes one dimension, and the Lie group that we are using is thus isomorphic to $SO(M_3)$. This is much smaller than the full Lie group used for the spatial representation [Pavlov et al. 2011] which had a dimension proportional to the square of the number of cells of the mesh—a potential reduction of several orders of magnitude.

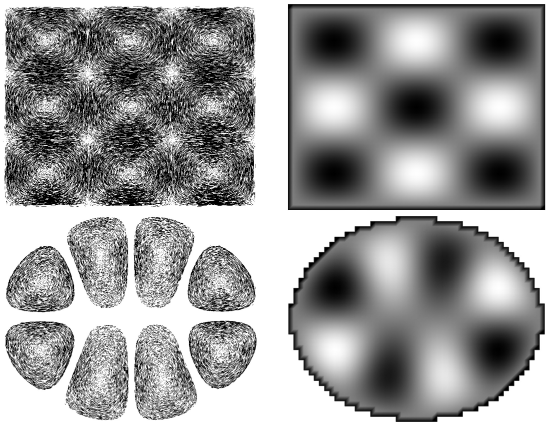


Figure 3: ***Effect of shape on spectral bases**: The Laplacian eigenvectors depends heavily on the domain Ω. Here, rectangle (top) vs. ellipse (bottom) domains (both computed on 2D rectangular grid of size 60^2) exhibit very different eigenvectors Ψ_{10} and Φ_{10}.*

Lie algebra. We identify each velocity eigenfunction Ψ_m with an element of the Lie algebra of the above Lie group as follows. We take the Lie derivative along the velocity field Ψ_m of a scalar eigenfunction Φ_j, then we project the resulting scalar field onto another scalar eigenfunction Φ_j, producing a matrix A_m for each velocity eigenfunction Ψ_m, with entries

$$A_{m,ij} = \int_{\mathcal{M}} \Phi_i (\Psi_m \cdot \nabla\Phi_j). \tag{2}$$

Computing A_m amounts to turning a 3-form into a dual 0-form first with $\star_3$, and then carrying out the integral in the (diamond) volume spanned by each face and its dual edge: this way, the differential of the dual 0-form from Φ_j is multiplied by the 2-form Ψ_m on the face and the average dual 0-form from Φ_i on the face. Notice that we have $\langle A_m, A_n \rangle = \delta_{mn}$ by construction thanks to the basis of Ψ being orthonormal. As in the non-spectral case, the divergence-free condition leads to the antisymmetry of these matrices since

$$A_{m,ij} + A_{m,ji} = \int_{\mathcal{M}} \Psi_m \cdot \nabla(\Phi_i \Phi_j) = -\int_{\mathcal{M}} \Phi_i \Phi_j \nabla \cdot \Psi_m = 0.$$

This is expected, since the Lie algebra $\mathfrak{so}(M_3)$ of $SO(M_3)$ contains only antisymmetric matrices. The Lie algebra has a Lie bracket operator, which is given by the usual matrix commutator $[A_m, A_n] = A_m A_n - A_n A_m$.

Non-holonomic constraint. Just like in [Pavlov et al. 2011], not every element of the Lie algebra $\mathfrak{so}(M_3)$ will correspond to a fluid velocity spanned by the eigenfunctions Ψ_m. We force the dynamics on the Lie algebra to remain within the domain of physically-sensible elements using the following non-holonomic constraint, which keeps the velocity within the space spanned by the lowest frequency $M_C + \beta_1$ divergence-free 2-form basis fields:

$$A = \sum_{i=1}^{M_C+\beta_1} v_i \, A_i \tag{3}$$

where v_i is a coefficient for A_i representing the modal amplitude of frequency κ_i. This linear condition can thus be seen as an intuitive extension of the one-away spatial constraint on Lie algebra elements used in [Pavlov et al. 2011] that we mentioned in §2.3.

3.3 Spectral variational integrator

The Lagrangian of the fluid motion (i.e., its kinetic energy in the case of Euler fluids) can be written as $\mathcal{L}_{\text{Euler}} = \frac{1}{2}\langle A(t), A(t)\rangle$ as we reviewed in §2.4. Thus, the equation of motion can be derived from Hamilton's (least action) principle

$$\int \langle A(t), \delta A\rangle dt = 0, \tag{4}$$

where $\delta A = \delta(\dot{q}q^{-1})$ is the variation of A induced by variation of q. If we denote $B \equiv \delta q q^{-1}$, one has $\delta A = \delta\dot{q}q^{-1} - \dot{q}q^{-1}\delta q q^{-1}$. Since $\dot{B} = \delta\dot{q}q^{-1} - \delta q q^{-1}\dot{q}q^{-1}$, we find that δA is induced by variations of q only if it satisfies Lin's constraints [Gawlik et al. 2011]:

$$\delta A = \dot{B} + [B, A], \tag{5}$$

where $B = \sum_i b_i A_i$ is an arbitrary element of the Lie algebra with coordinates $\{b_i\}_i$ in the 2-form basis. Substituting Eq. (5) into Eq. (4), we then have

$$\begin{aligned} 0 &= \int \langle A, \delta A\rangle \, dt \\ &= \int \sum_{i,k} v_i \dot{b}_k \, \langle A_i, A_k\rangle + \sum_{i,j,k} v_i v_j b_k \, \langle A_i, [A_k, A_j]\rangle \, dt \\ &= \int \sum_k \Big(-\sum_i \dot{v}_i \, \langle A_i, A_k\rangle + \sum_{i,j} v_i v_j \, \langle A_i, [A_k, A_j]\rangle \Big) \, b_k \, dt. \end{aligned}$$

Since this last equation must be valid for any b_k, the update rule for the velocity field has to be:

$$\dot{v}_k = \sum_i \dot{v}_i \, \langle A_i, A_k\rangle = \sum_{i,j} v_i v_j \, \langle A_i, [A_k, A_j]\rangle \equiv \mathbf{v}^t \mathbf{C}_k \mathbf{v}, \tag{6}$$

where $\mathbf{v}$ is the column vector storing the coefficients v_i of the discrete velocity A (Eq. (3)), and $\mathbf{C}_k$ is the square matrix with

$$C_{k,ij} = \langle A_i, [A_k, A_j]\rangle = \int_{\mathcal{M}} (\nabla \times \Psi_i) \cdot (\Psi_k \times \Psi_j). \tag{7}$$

Note that this velocity update do not even require the scalar (3-form) bases used in the definition of the Lie group; however, these bases become important in more general simulations, including magnetohydrodynamics and rotating stratified flows.

Time integrator. The continuous-time update in Eq. (6) is then discretized via either a midpoint rule (which will lead to an *energy-preserving* model-reduced variant of [Mullen et al. 2009]) or a trapezoidal rule (which corresponds to a model-reduced variant of the variational method of [Pavlov et al. 2011]). Specifically, the midpoint rule is implemented as

$$v_k^{t+h} - v_k^t = h \sum_{i,j} C_{k,ij} \frac{v_i^t + v_i^{t+h}}{2} \frac{v_j^t + v_j^{t+h}}{2}, \tag{8}$$

The energy preservation can be easily verified by multiplying $v_k^t + v_k^{t+h}$ on both sides of the above equation, summing over k, and invoking the property of coefficients $C_{k,ij} = -C_{j,ik}$. The trapezoidal rule can, instead, be implemented as

$$v_k^{t+h} - v_k^t = \frac{h}{2} \sum_{i,j} C_{k,ij} (v_i^t v_j^t + v_i^{t+h} v_j^{t+h}), \tag{9}$$

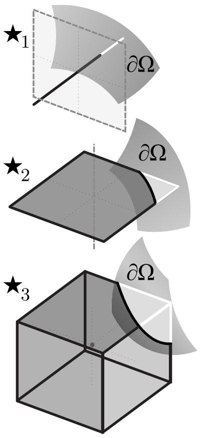
which is derived from a temporal discretization of the action with variation of $(\delta q)q^{-1}$ for q along the path to be in the restricted Lie algebra set (to enforce Lin constraints), see App. D. Both the energy-preserving and trapezoidal variational rules are time-reversible implicit methods solved through a simple quadratic set of equations with a small number of variables. An explicit forward Euler integration can also be used for small time steps in order to further reduce computational complexity; no guarantee of good behavior over long periods of time is available in this case.

Discussion. Our structural coefficients $\mathbf{C}_k$ (which can be precomputed once the spectral bases are found) are similar to the advection terms mentioned in [De Witt et al. 2012]. However, there are some important differences. Although both expressions converge to the same continuous limit, our variational approach produces coefficients that are exactly antisymmetric in j and k as the Lie bracket is anti-symmetric, making our method energy-preserving *without* the artifact-prone energy renormalization step advocated in their work. We also note that the symmetry mentioned in their discretization (specifically, $\kappa_j^2 C_{k,ij} = -\kappa_i^2 C_{k,ji}$) is, in fact, *only* valid in 2D as we explain in §B. Moreover, our variational integrator also admits a spectral version of Kelvin's theorem as detailed in §C. Finally, our approach is quite different from Azencot et al. [2014] even though they, too, use an operator representation of vector fields. Because they explicitly leverage the scalar nature of vorticity in 2D, their work cannot be generalized to 3D. Additionally, their representation of vector (resp., vorticity) fields relies on spatial, piecewise-linear (resp., piecewise constant) basis functions instead of using a reduced set of basis functions.

3.4 Embedding complex domains on Cartesian grids

Unstructured meshes can be made to conform to arbitrary domains, and the construction of Laplacians on simplicial meshes is well documented (see §A). Therefore, one could use our approach on simplicial meshes directly (see Fig. 4 for an example on a non-flat triangle mesh). However, Cartesian (regular) grids always generate *much simpler* data structures and *sparser stencils* for the structural coefficients, so sticking to Cartesian grids is key when efficiency is paramount. Yet, model-reduced fluid methods cannot easily deal with complex domains using only a regular grid to embed it in.

We propose a simple extension of [Ng et al. 2009] to compute k-form Laplacians of an arbitrary domain, *still using a regular grid*. This renders the implementation of Laplacians and their boundary conditions quite trivial, and removes the arduous task of tetrahedralizing arbitrary domains. This idea was introduced in [Batty et al. 2007] for their pressure-based projection, and a simple alteration proposed by [Ng et al. 2009] made the approach robust and convergent. We leverage this latter work by noticing that the modification of the Laplacian Δ_3 that they proposed amounts to a local change to the Hodge star operator $\star_2$.

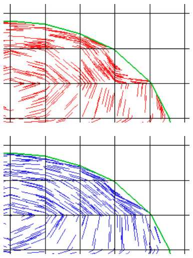


More precisely, consider a domain Ω, e.g. defined implicitly by a function χ via $\Omega = \{\mathbf{x} \,|\, \chi(\mathbf{x}) \geq 0\}$. Recall that the diagonal Hodge stars on a mesh $\mathcal{M}$ are all expressed using local ratios of measurements (edge lengths, face areas, cell volumes) on both the primal elements of $\mathcal{M}$ and its dual elements [Desbrun et al. 2008]. The changes to the Laplacian operator Δ_3 that Ng et al. [2009] introduced can be reexpressed by an alteration of the Hodge star $\star_2$ where *each primal area measurement only counts the part of the primal face that is inside* Ω, but dual edge lengths are kept unchanged. We extend this simple observation (which amounts to a local, numerical homogenization to capture sub-grid resolution) by computing modified Hodge stars $\hat{\star}_1$, $\hat{\star}_2$, and $\hat{\star}_3$ where only the parts of the primal elements (partial lengths, areas, or volumes) that are within the domain Ω are counted (see inset). Note that changing directly the Hodge stars does not affect the symmetry and positive-definiteness of the Laplacians, and thus incurs no additional cost for our method.

This straightforward extension allows us to compute our spectral bases on regular grid for arbitrary domains Ω as illustrated in the

Algorithm 1 Model-reduced variational integrator

```
Construct 2-form spectral basis with selected frequencies
if scalar fields needed then
    Construct 3-form basis with same selected frequencies
end if
Construct structural coefficients $C_{k,ij}$ with Eq. (7)
Initialize simulation
for each time step do
    Time integration through explicit update or Eq. (8)/Eq. (9)
    If needed, perform scalar advection in current velocity field
end for
```



inset (comparing, on a coarse grid for clarity, an eigenvector field using a voxelized approximation of the boundary (top) vs. Hodge modification (bottom)) and in Fig. 5 for a basis element of vector fields. We also show the behavior of this Hodge star modification under refinement of the regular grid for a given continuous elliptic domain Ω, resulting in very good approximations of the eigenvectors. Note that the Hodge star operators may involve division by small denominators. A simple and typical thresholding of numbers below the average precision of the floating point representation (i.e., 1e-8) to avoid division by zero is enough, and the eigenfunctions with our specific tangential and normal boundary conditions are computed without any extra preprocessing. In fact, large values of the Hodge stars act as penalty: if a small fraction of the face is inside the domain, then only an accordingly-small flux is allowed.

3.5 Variants and extensions

Our approach has been limited to Euler equations so far. However, the use of spectral basis functions, the ability to deal with arbitrary domains on a regular grid, and the functional-map nature of the discretization makes for a very versatile framework in which extensions to the Euler fluid model can be easily incorporated. In all cases, the pseudocode remains identical, as outlined in Algorithm 1.

Viscosity. While the ability to remove spurious energy dissipation is important for the consistency of a numerical integrator with respect to time step size, real fluids exhibit viscosity. Adding viscosity is easily achieved: it corresponds to a dissipation of modal amplitudes by a factor of κ_i^2 since the vector-valued Laplacian is a diagonal matrix in the spectral domain as already leveraged in, e.g., [Stam 2002]. Thus each modal magnitude v_i evolves as

$$\dot{v}_i = -\nu\kappa_i^2 v_i$$

for a viscosity coefficient of ν. Consequently, Navier-Stokes equations can be handled through an operator-splitting approach by updating the modal magnitudes over each time interval h through

$$v_i^{t+h} \leftarrow e^{-\nu\kappa_i^2 h} v_i^t.$$

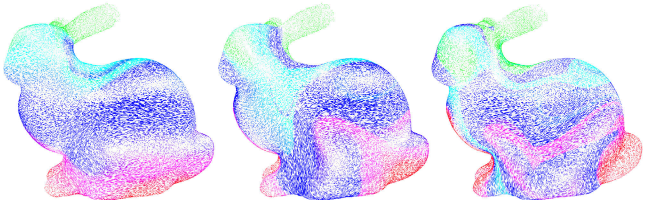


Figure 4: ***Curved domain**: While all other figures were achieved on a regular grid, our approach applies to arbitrary domains, here on the* surface *of a triangulated domain; a simple laminar flow with initial horizontal velocity smoothly varying along the vertical direction quickly develops vortical structures on this complex surface.*

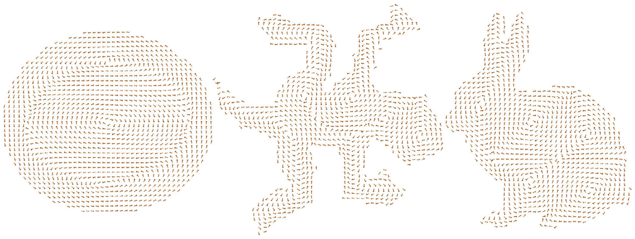


Figure 5: ***Domain-altered Hodge stars**: Our framework can generate vector bases satisfying prescribed boundary condition for arbitrary domains embedded in a Cartesian grid. Hedge-hog visualization of Ψ_5 on a 256^2 grid for three different 2D domain shapes, obtained through a simple alteration of the Hodge star $\star$ operator.*

Magnetohydrodynamics (MHD). The equations for ideal, incompressible MHD are easily expressed as a modification of the Euler equations. The Lagrangian of MHD is

$$\mathcal{L}_{\text{MHD}} = \frac{1}{2}\langle \mathbf{v}, \mathbf{v}\rangle - \frac{1}{2}\langle \mathbf{F}, \mathbf{F}\rangle$$

in appropriate units [Gawlik et al. 2011], where $\mathbf{F}$ is the magnetic field which is advected by the velocity field $\mathbf{v}$. As both $\mathbf{v}$ and $\mathbf{F}$ are divergence-free, they can be discretized with our divergence-free spectral bases. With such discrete velocity and magnetic fields, the update rule in time to simulate the MHD equations closely resembles the Euler fluid case:

$$\dot{v}_k = \mathbf{v}^t \mathbf{C}_k \mathbf{v} - \mathbf{F}^t \mathbf{C}_k \mathbf{F},$$
$$\dot{F}_k = \sum\nolimits_{ij} F_i v_j C_{j,ki},$$

where the second equation performs the advection of $\mathbf{F}$ (encoded in our spectral 2-form basis) in the current velocity v. It is easy to verify that the cross helicity $\int_{\mathcal{M}} \mathbf{v} \cdot \mathbf{F} = \sum_i v_i F_i$ is exactly preserved in our integrator as it is in the continuous world—a numerical property rarely satisfied in existing integrators [Gawlik et al. 2011]. The cross helicity is related to the topological linking of the magnetic field and the fluid vorticity. Thus, its preservation prevents spurious changes in the topology of the magnetic field lines over the course of a simulation. This extension to MHD can also be applied to a series of other Euler-Poincaré equations with advected parameters, modeling nematic liquid crystal flows and microstretch continua among others [Gawlik et al. 2011]. Note that buoyancy and density fields can be handled in a similar fashion, whether they are coupled with the dynamics or passively advected; they just need to be smooth enough to be captured by low frequencies. Our framework can therefore be used for, e.g., atmospheric simulations as well [Desbrun et al. 2013].

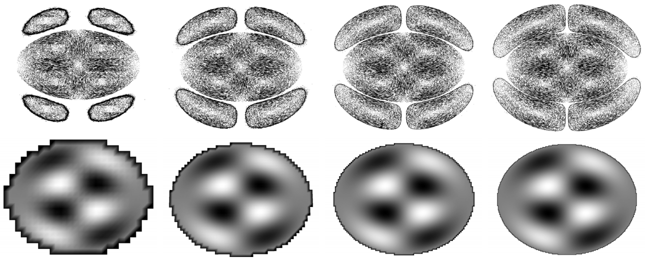


Figure 6: ***Convergence of Laplacians**: Our discretization of the two Laplacians creates (vector and scalar) eigenfields that converge under refinement of the regular grid used to compute them, extending the linear convergence proved in [Ng et al. 2009]. Here, particle-tracing visualization of the 15^{th} eigenbasis for vector fields on the ellipse (top) at resolution 30^2, 60^2, 120^2, and 240^2, and 15^{th} eigen function (bottom) at the same resolutions.*

Subgrid scale modeling. Direct numerical solvers using Navier Stokes equation require high resolutions to resolve the correct coupling of large and small scale structures. Instead, subgrid scale modeling requires much fewer degrees of freedom to capture the correct large-scale structures by simulating the main effects of the small subgrid scale structures without actually resolving them. Among the many models that match empirical data well, the LANS-α model (Lagrangian-Averaged-Navier-Stokes, see [Foias et al. 2002]) advects the velocity in a filtered velocity to better capture the correct energy cascading. Since filtering is achieved through a Laplacian-based Helmholtz operator, we can also use our spectral approach to simulate this model with ease. The kinetic energy (i.e., Lagrangian) is now defined as

$$\mathcal{L}_{\alpha\text{-Euler}} = \int_{\mathcal{M}} \frac{1}{2}\mathbf{v}^2 + \frac{\alpha^2}{2}|\nabla\mathbf{v}|^2.$$

In our spectral bases, the α-model amounts to adding kinetic energy terms scaled by κ_i^2 for the modal amplitudes v_i. Hamilton's principle for the modified Lagrangian leads to essentially the same update rule as Navier-Stokes', except that the structural coefficients are replaced by $(1+\alpha^2\kappa_i^2)C_{k,ij}$. Note that this modification keeps the antisymmetry in j and k intact, and is therefore a trivial alteration of the basic scheme.

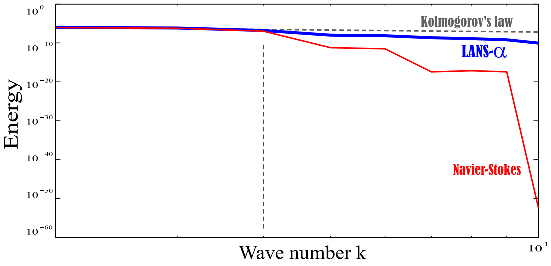


Figure 7: ***Spectral energy distribution**: With forcing terms keeping the low wave number amplitudes fixed [Foias et al. 2002], our 3D reduced model applied to the LANS-α model of turbulence produces an average spectral energy distribution (blue) much closer to the expected Kolmogorov distribution (black) than with the usual Navier-Stokes equations (red).*

Moving obstacles and external forces. For moving obstacles, instead of calculating additional boundary bases as in [Treuille et al. 2006], we instead follow the simpler solution proposed in [De Witt et al. 2012]: the difference in the normal component of the velocity on the boundary is projected onto the velocity basis and subtracted from $\mathbf{v}$, resulting in a low-frequency field roughly satisfying the boundary condition. For any external forces, e.g., buoyancy forces, their effects on the time derivative of the modal amplitude v_i of the i-th frequency are simply calculated by their projection onto Ψ_i. The results are visually correct even on complex shapes, and with minimal computational overhead (see Fig. 10).

3.6 Generalization to other bases

While we provided detail on the construction of a variational model-reduced integrator for fluid simulation using Laplace eigenvectors, one can easily adapt our approach to arbitrary basis functions, even those extracted from a training set of fluid motions. Suppose that we are given a set of scalar basis elements Φ_i (orthonormalized through the Gram-Schmidt procedure) and a set of velocity basis elements Ψ_i. The Lie derivative matrix A will still be antisymmetric as long as Ψ_i's are divergence-free. This means that one can use existing finite element basis functions instead of our Laplace eigenvectors—or even wavelet bases of $H(\text{div}, \Omega)$ (see, e.g., [Urban 2002]) if spatially localized basis functions are sought after to get a sparser advection. The key to the numerical benefits of our variational approach is to ensure the anti-commutativity of the Lie bracket in the evaluation of $\langle A_i, [A_j, A_k]\rangle$ (Eq. (7)) and the advection of other fields by the exponential map (or approximations thereof, see App. D) of the matrix representing the Lie derivative as done in MHD and complex fluids [Gawlik et al. 2011]. In a way, the original non-spectral variational integrators can be seen as a special case of our framework where Whitney basis functions are used. However, viscosity can no longer be handled as easily in this case as the Laplacian is not diagonal in general bases. Moreover, the required number of degrees of freedom to produce smooth flows may end up being high if the bases are arbitrary.

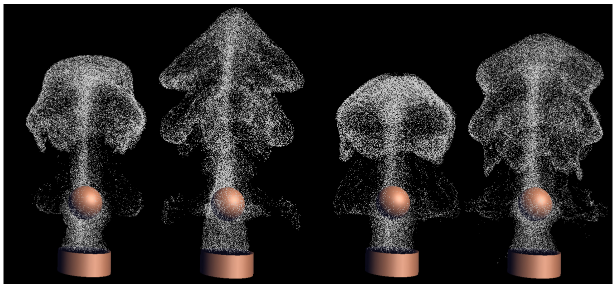


Figure 8: ***Smoke rising.** Using only 230 modes (about 0.003% of the full spectrum simulation), both [De Witt et al. 2012]'s (left) and our approach already exhibit the expected volutes for a buoyancy-driven flow over a sphere.*

4 Results

Our results were generated on an Intel i7 laptop with 12GB RAM, and visualized using our own particle-tracing and rendering tools.

Reduced vs. full simulation. In order to check the validity of our reduced approach, we performed a stress test in a periodic 2D domain to visualize how the increase in the number of bases used in our spectral integrator impacts the simulation over time. We selected a band-limited initial velocity field at time $t = 0$ that only contains non-zero components for the first 120 frequencies. We then advected the fluid using our integrator, with fluid markers initially set as two colored disks near the center. Because of the propensity of vorticity to go to higher scales, our reduced approach does not lead to the exact same position of the fluid markers after 12s of simulation if one uses only 120 bases. However, as the number of bases increases to 300 or 500, the simulation quickly captures the same dynamical behavior as a full variational integrator with 256^2 degrees of freedom, see Fig. 9. We demonstrate in the accompanying video that our variational integrator also captures the proper qualitative behavior of two merging vortices in contrast to the result using [De Witt et al. 2012] which, instead, generates several vortices (see supplementary video).

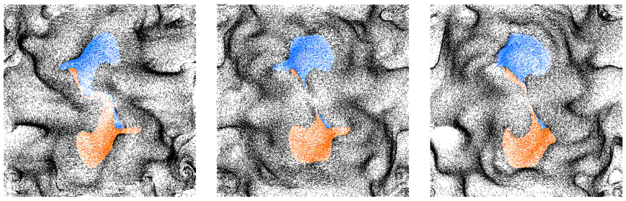


Figure 9: ***Convergence of simulation**: A flow in a periodic domain is initialized with a band-limited velocity fields with 120 wave number vectors. Fluid markers (forming a blue and red circle) are added for visualization. After 12s of simulation, the results of our reduced approach (left: 120, middle: 300 modes) vs. the full 256^2 dynamics (right) are qualitatively similar.*

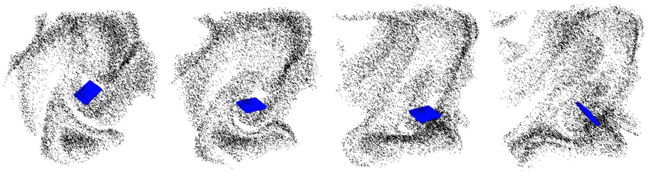
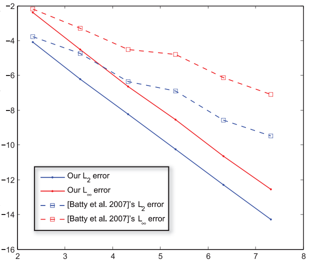


Figure 10: ***Interactivity**: We can also use the analytic expressions for Ψ_k and $C_{k,ij}$ in a periodic 3D domain to handle a large number of modes directly. The explicit update rule exhibits no artificial damping of the energy as expected, but offers realtime flows.*

Arbitrary domains. We also show in Figs. 3, 5, and 11 (2D) and Figs. 1, 2, 8, 14, and 10 (3D) that the use of boundary conditions embedded on regular grids leads to the expected visual behavior near domain boundaries, eliminating the staircase artifacts of traditional immersed-grid methods. Our homogenized boundary treatment obtains results similar to those of unstructured meshes while using only calculations that are directly performed on regular grids—thus requiring significantly simpler, smaller, and more efficient data structures. As shown in the log(error)-log(resolution) plot in the inset, our basis fields converge with second order accuracy, much faster than the boundary condition in [Batty et al. 2007] (the latter may, in fact, not even converge in some cases as discussed in [Ng et al. 2009]). Moreover, we also extended this approach not just to scalar field, but vector fields. Our fluid dynamics is also consistent across a wide range of temporal and spatial discretizations, see Fig. 11. In addition, the regular grid structure also simplifies the interaction with immersed solid objects as demonstrated in Fig. 14 through a flow induced by a scripted car turning around a corner; interactive fluid stirring by a paddle manipulated by the user is also easily achieved as shown in Fig. 10. We also show in Fig. 8 that our method can handle the typical test case of smoke plume past a sphere even at low resolution, and we can incorporate both free-slip or no-slip boundary conditions. Finally, our spectral integrator can be carried out in the same fashion on curved domains as well, since the eigenvectors of the Laplace(-Beltrami) operator are no more difficult to compute on a triangulated surface; Fig. 4 shows a simple laminar flow on the *surface* of the bunny model.



Advanced fluid models. We also extended our method to the LANS-α turbulence model to better capture the spectral energy distribution with a small number of modes. On a 3D regular grid, we performed a simulation as described in [Foias et al. 2002] by holding the low wave number components v_i fixed for $|\kappa_i| < 2$ to act as a forcing term, and running the simulation until $t = 100$. We then extracted the average spectral energy distribution present between $t = 33$ to $t = 100$. We show in Fig. 7 that the Kolmogorov "$-5/3$ law" is much better captured than with the usual Navier-Stokes model, even for the low number of modes used in our spectral context: the α-model produces a decay rate at high wave numbers much steeper than a Navier-Stokes simulation, allowing us to cut off the higher frequencies at a lower threshold without significant deviation from the spectral distribution. This indicates that our approach consisting in a simple scaling of the structural coefficients helps improving fluid simulation on coarse grids.

We also implemented our extension to MHD, and found the expected preservation of cross-helicity and energy. For comparison purposes (see, e.g., [Gawlik et al. 2011]), we visualize our results of the typical rotor test with 100 modes in Fig. 15. We finally show in Figs. 2 and 8 that buoyancy forces are also easy to incorporate by adding an upwards force proportional to the local smoke temperature; the curl of this external force is projected onto the 1-form basis functions, and used to update the vorticity.

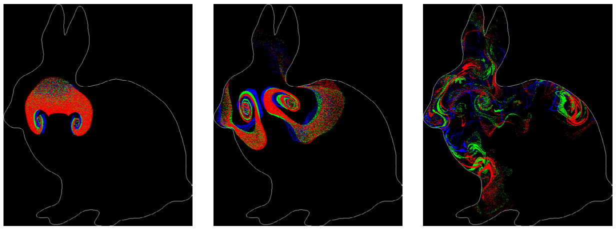


Figure 11: ***Robustness to resolution**: With the homogenized boundary condition on grids of resolution 40^2 (blue), 80^2 (green), and 160^2 (red), no staircase artifacts are observed, and the simulation results are consistent across resolutions.*

Computational efficiency. Our use of model reduction via Laplacian eigenbases provides a significantly more efficient alternative to full simulators, obviously. Due to our variational treatment of time integration, we also prevent many shortcomings of the previous reduced models as we ensure consistency of the results over a large spectrum of spatial and temporal discretization rates, and maintain a qualitatively correct behavior even on coarse grids. The efficiency gain compared to the full variational simulation is apparent, be it in 2D, curved 2D, or 3D. For instance, a full-blown 128^2 grid takes around 50s for the variational integrator to update one step (through a Newton solver) in a typical simulation using the trapezoidal rule update, while a 50-mode (resp., 100-mode and 200-mode) simulation with our integrator takes only 0.098s (resp., 0.65s and 2.0s) for complex boundaries (i.e., with dense structural coefficients), and 0.026s (resp., 0.070s, 0.28s) for simple box domains (with sparse coefficients). Our Newton solver normally converges in a couple of iterations depending on the time step size (which determines the quality of the initial guess); for instance, the average in our 3D bunny buoyancy test in Fig. 2 is below 3 iterations.

Selection of frequencies. Depending on how many modes the user is willing to discard (and replace by wavelet noise or dynamical texture for efficiency), the computational gains can be in the order of several orders of magnitude, and this allows us to simulate flows at interactive or realtime rates (see Fig. 14, or Fig. 10 for an example with a periodic 3D domain where we can compute the eigenbases in closed form). Note however that our model-reduced integrator suffers from the usual limitation of model reduction: the complexity is actually growing quadratically (resp., cubically) with the number of modes for sparse (resp., dense) structural coefficients. So our integrator is numerically efficient only for relatively low mode counts. However, this is exactly the regime for which one can achieve significant computational savings for very little visual degradation. Similar to [De Witt et al. 2012], we also found that, when using very few low frequencies leads to unappealing simulations, adding a few high frequencies (and thus, skipping a large amount of medium frequencies) is enough to render an animation realistic: our integrator can use such a tailored frequency range seamlessly, and the non-linear exchange between low and high frequencies is enough to create much more complex patterns that respect the expected motion of the flow, see Fig. 12. Fig 8 was also done in this manner: the first 115 modes were used, the next 400 modes were skipped, and we added the next 115 modes to add small scale effects. The use of subgrid scale modeling explained in §3.5 is yet another way to make sure that the higher frequencies are properly dealt with and provide a good, visually correct approximation to the fluid equations.

Time stepping. Finally, an important feature of our model-reduced approach is its ability to handle both explicit and implicit integration. Implicit integration, using the midpoint rule (Eq. (8)) or the trapezoidal rule (Eq. (9)), come with good numerical guarantees due to the time reversibility. However, explicit integration is also very convenient as it further reduces the time complexity of the simulation. Nevertheless, an explicit integration has very little theoretical guarantees, and should only be used with care.

Quantitative experiments. One way to evaluate a reduced model of fluids is to measure the evolution in time of the error of the velocity field compared to a full-spectrum (spatial) simulation. Using the exact same Laplacian eigenvectors representing only 0.003% of the modes for the 3D simulation of the rising smoke in Fig. 8, both our structural coefficients and [De Witt et al. 2012]'s provide a slowly increasing error as shown in Fig. 13 (top), with ours showing an improvement of around 20%; the L_∞ shows a more pronounced improvement as well. The same experiment in 2D for 2 vortices exhibits the same trend (Fig. 13, bottom), with a more pronounced difference given that our approach leads to the two vortices merging as in the ground truth simulation, while [De Witt et al. 2012]'s generates multiple vortices (see supplementary video).

5 Conclusion

We have introduced a variational integrator for fluid simulation in reduced coordinates. By restricting the variations in Hamilton's principle to a low-dimensional space spanned by low-frequency divergence-free velocity fields, our method exhibits the properties of variational integrators in capturing the qualitatively correct behavior of ideal incompressible fluids (such as Kelvin's circulation and energy preservation) while greatly reducing the computational cost. We further facilitated its use in graphics applications by providing a method to calculate the necessary bases for arbitrary domains embedded in a regular grid through a simple and novel alteration to the Hodge star operators at the domain's boundary to offer sub-grid accuracy at no extra cost. Finally, we demonstrated the versatility of our integrator by straightforward extensions to moving boundary, magnetohydrodynamics, and turbulence models.

Discussion. Algorithmically, our method resembles all other model-reduced fluid methods. However, it offers a *unified* formulation of model-reduced fluid flows for arbitrary basis functions, and provides structural coefficients without the artifacts of [De Witt et al. 2012]. It only requires a regular grid to encode arbitrary domains with the same convergence rate as [Ng et al. 2009], but also allows for the computation of vorticity bases. DEC operators for the computation of Laplacians allow full control over boundary conditions, for arbitrary topology; but any other discrete operators can easily be used instead. Our nonlinear update rules enforce a discrete form of Kelvin's theorem and time reversibility—and thus energy preservation. They also offer robustness to time and space discretization rates, an important feature when previewing results.

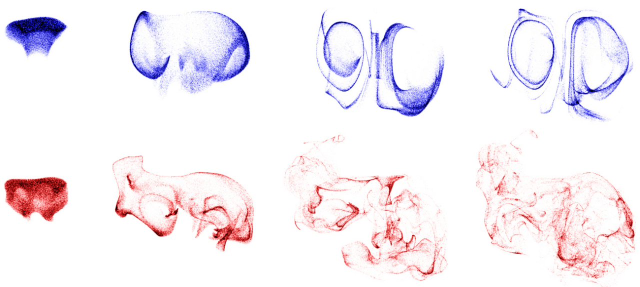


Figure 12: ***Frequency shaping**: For the same setup as Fig. 2, using only the lowest 10 eigenbasis functions for vector fields leads to a very limited motion. However, adding another 10 basis functions of high frequencies creates a much more detailed animation at very little cost, instead of using all the frequencies from low to high.*

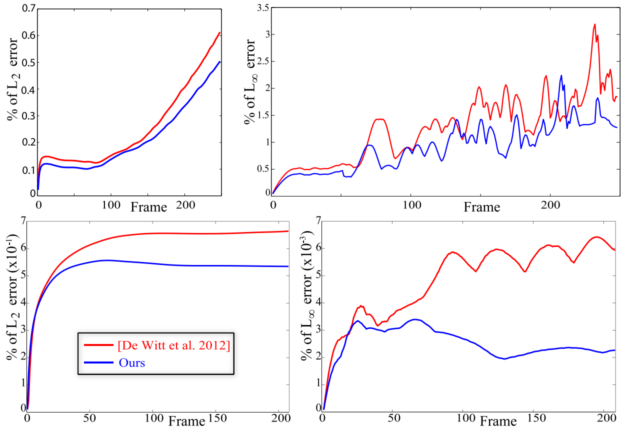


Figure 13: ***Relative errors.** Relative L_2 (left) and L_∞ errors measured with respect to a full-spectrum (spatial) simulation are systematically improved with our structural coefficients compared to [De Witt et al. 2012], even if the same time integration is used to allow for a fair comparison. Top: errors for the rising smoke example of Fig. 8; bottom: errors for two merging vortices (see video).*

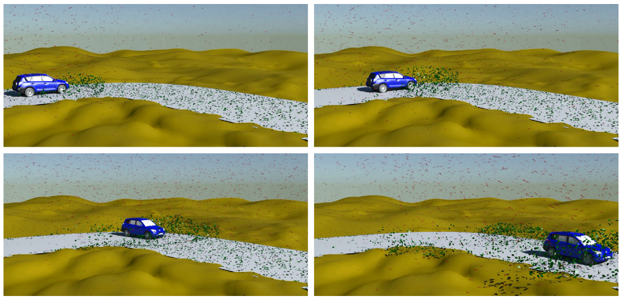


Figure 14: ***Immersed moving objects.** As the car makes a right turn, the low frequency motion of the air displaced around it lifts the dead leaves. The velocity field above is visualized through arrows.*

One can also employ explicit integration to further reduce computational complexity. Any application seeking low time complexity of fluid simulation will benefit from our reduced space approach, in particular, real-time interactive simulation or tools for designing artist-driven coarse simulation. Even production-quality smoke or fluid animation may be achieved without having recourse to a full resolution simulation through existing post-process curl noise techniques. Moreover, our modified Hodge star can be used in a variety of geometry processing applications where subgrid accuracy on coarse grids is desirable.

Future work. One intriguing extension is the use of reduced bases with spatial locality, as our integrator is not restricted to any particular set of bases functions. For instance, using wavelets for vorticity may offer *optimal sparsity in the structural coefficients* if we can address the challenge of adapting the frequency to the local feature size of the domain. Improving scalability (through sparsity in structural coefficients or pseudo-spectral methods) and better adaptivity to moving/deforming solid boundaries (through spatial locality) may then offer a wider applicability for model reduced methods. Another possible future extension is to incorporate free surface boundary conditions through our modified Hodge star, combined with wavelet representations for the volume of fluid per cell.

Acknowledgements

Partial funding was provided by NSF grants CCF-1011944, CMMI-1250261, III-1302285 and IIS-0953096. MD gratefully acknowledges all the members of the TITANE team and the Inria International Chair program for support.

References

An, S., Kim, T., and James, D. 2008. Optimizing cubature for efficient integration of subspace deformations. *ACM Trans. Graph. 27*, 5, Art. 165.

Arnold, D. N., Falk, R. S., and Winther, R. 2006. Finite element exterior calculus, homological techniques, and applications. *Acta Numerica 15*, 1–155.

Azencot, O., Weissmann, S., Ovsjanikov, M., Wardetzky, M., and Ben-Chen, M. 2014. Functional fluids on surfaces. *Comput. Graph. Forum 33*, 5, 237–246.

Batty, C., Bertails, F., and Bridson, R. 2007. A fast variational framework for accurate solid-fluid coupling. *ACM Trans. Graph. 26*, 3, Art. 100.

Bell, N., and Hirani, A. N., 2008. PyDEC: A Python library for Discrete Exterior Calculus. Google Code project at: `http://code.google.com/p/pydec/`.

Chorin, A., and Marsden, J. 1979. *A Mathematical Introduction to Fluid Mechanics*, 3rd edition ed. Springer-Verlag.

Cohen, J. M., Tariq, S., and Green, S. 2010. Interactive fluid-particle simulation using translating Eulerian grids. In *ACM Symp. on Interactive 3D Graphics and Games*, 15–22.

De Witt, T., Lessig, C., and Fiume, E. 2012. Fluid simulation using Laplacian eigenfunctions. *ACM Trans. Graph. 31*, 1, Art. 10.

Desbrun, M., Kanso, E., and Tong, Y. 2008. Discrete differential forms for computational modeling. In *Discrete Differential Geometry*, A. I. Bobenko et al., Ed., vol. 38 of *Oberwolfach Seminars*. Birkhäuser Basel, 287–324.

Desbrun, M., Gawlik, E. S., Gay-Balmaz, F., and Zeitlin, V. 2013. Variational discretization for rotating stratified fluids. *Disc. Cont. Dyn. S. 34*, 2, 477–509.

Elcott, S., and Schröder, P. 2006. Building your own DEC at home. In *Discrete Differential Geometry*, ACM SIGGRAPH Courses, 55–59.

Elcott, S., Tong, Y., Kanso, E., Schröder, P., and Desbrun, M. 2007. Stable, circulation-preserving, simplicial fluids. *ACM Trans. Graph. 26*, 1, Art. 4.

Feldman, B. E., O'Brien, J. F., and Klingner, B. M. 2005. Animating gases with hybrid meshes. *ACM Trans. Graph. 24*, 3, 904–909.

Foias, C., Holm, D. D., and Titi, E. S. 2002. The three dimensional viscous Camassa–Holm equations, and their relation to the Navier–Stokes equations and turbulence theory. *J. Dyn. Differ. Equ. 14*, 1, 1–35.

Foster, N., and Metaxas, D. 1997. Modeling the motion of a hot, turbulent gas. In *Proc. ACM SIGGRAPH*, 181–188.

Gao, Y., Li, C.-F., Ren, B., and Hu, S.-M. 2013. View-dependent multiscale fluid simulation. *IEEE Trans. Vis. Comput. Graph. 19*, 2, 178–188.

Gawlik, E., Mullen, P., Pavlov, D., Marsden, J., and Desbrun, M. 2011. Geometric, variational discretization of continuum theories. *Physica D: Nonlinear Phenomena 240*, 21, 1724–1760.

Golas, A., Narain, R., Sewall, J., Krajcevski, P., Dubey, P., and Lin, M. 2012. Large-scale fluid simulation using velocity-vorticity domain decomposition. *ACM Trans. Graph. 31*, 6, Art. 148.

Gupta, M., and Narasimhan, S. G. 2007. Legendre fluids: A unified framework for analytic reduced space modeling and rendering of participating media. In *Symp. on Computer Animation*, 17–25.

Harlow, F. H., and Welch, J. E. 1965. Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. *Physics of Fluids 8*, 12, 2182–2189.

Howes, R., Schroeder, C., and Teran, J. M. 2013. A virtual node algorithm for Hodge decompositions of inviscid flow problems with irregular domains. *Methods Appl. Anal. 20*, 4, 439–455.

Kharevych, L., Yang, W., Tong, Y., Kanso, E., Marsden, J. E., Schröder, P., and Desbrun, M. 2006. Geometric, variational integrators for computer animation. In *Symp. on Computer Animation*, 43–51.

Kim, T., and Delaney, J. 2013. Subspace fluid re-simulation. *ACM Trans. Graph. 32*, 4, 62.

Kim, T., Thürey, N., James, D., and Gross, M. 2008. Wavelet turbulence for fluid simulation. In *ACM Trans. Graph.*, vol. 27, ACM, 50.

Li, S., Huang, J., de Goes, F., Jin, X., Bao, H., and Desbrun, M. 2014. Space-time editing of elastic motion through material optimization and reduction. *ACM Trans. Graph. 33*, 4, Art. 108.

Long, B., and Reinhard, E. 2009. Real-time fluid simulation using discrete sine/cosine transforms. In *Symp. on Interactive 3D Graphics and Games*, 99–106.

Losasso, F., Talton, J., Kwatra, N., and Fedkiw, R. 2008. Two-way coupled SPH and Particle Level Set fluid simulation. *IEEE Trans. Vis. Comput. Graph. 14*, 4, 797–804.

Marsden, J. E., and West, M. 2001. Discrete mechanics and variational integrators. *Acta Numerica 2001 10*, 357–514.

Mullen, P., Crane, K., Pavlov, D., Tong, Y., and Desbrun, M. 2009. Energy-preserving integrators for fluid animation. *ACM Trans. Graph. 28*, 3, Art. 38.

Narain, R., Sewall, J., Carlson, M., and Lin, M. C. 2008. Fast animation of turbulence using energy transport and procedural synthesis. *ACM Trans. Graph. 27*, 5, 166.

Ng, Y. T., Min, C., and Gibou, F. 2009. An efficient fluid-solid coupling algorithm for single-phase flows. *J. Comput. Phys. 228*, 23, 8807–8829.

Orszag, S. A. 1969. Numerical methods for the simulation of turbulence. *Physics of Fluids 12*, II 250–257.

Pavlov, D., Mullen, P., Tong, Y., Kanso, E., Marsden, J., and Desbrun, M. 2011. Structure-preserving discretization of incompressible fluids. *Physica D: Nonlinear Phenomena 240*, 6, 443–458.

Schechter, H., and Bridson, R. 2008. Evolving sub-grid turbulence for smoke animation. In *Symp. on Computer Animation*, 1–7.

SELLE, A., RASMUSSEN, N., AND FEDKIW, R. 2005. A vortex particle method for smoke, water and explosions. *ACM Trans. Graph. 24*, 3, 910–914.

SILBERMAN, I. 1954. Planetary waves in the atmosphere. *J. Meteor. 11*, 27–34.

STAM, J., AND FIUME, E. 1993. Turbulent wind fields for gaseous phenomena. In *Proc. ACM SIGGRAPH*, 369–376.

STAM, J. 1999. Stable fluids. In *Proc. ACM SIGGRAPH*, 121–128.

STAM, J. 2002. A simple fluid solver based on the FFT. *J. Graph. Tools 6*, 2, 43–52.

STAM, J. 2003. Flows on surfaces of arbitrary topology. *ACM Trans. Graph. 22*, 3, 724–731.

STANTON, M., SHENG, Y., WICKE, M., PERAZZI, F., YUEN, A., NARASIMHAN, S., AND TREUILLE, A. 2013. Non-polynomial Galerkin projection on deforming meshes. *ACM Trans. Graph. 32*, 4, Art. 86.

STEINHOFF, J., AND UNDERHILL, D. 1994. Modification of the Euler equations for Vorticity Confinement. *Physics of Fluids 6*, 8, 2738–2744.

TONG, Y., ALLIEZ, P., COHEN-STEINER, D., AND DESBRUN, M. 2006. Designing quadrangulations with discrete harmonic forms. In *Symp. on Geometry Processing*, 201–210.

TREUILLE, A., LEWIS, A., AND POPOVIĆ, Z. 2006. Model reduction for real-time fluids. *ACM Trans. Graph. 25*, 3, 826–834.

URBAN, K. 2002. Wavelet bases for H(div) and H(curl). In *Wavelets in Numerical Simulation*, vol. 22 of *Lecture Notes in Computational Science and Engineering*. 83–107.

VON TYCOWICZ, C., SCHULZ, C., SEIDEL, H.-P., AND HILDEBRANDT, K. 2013. An efficient construction of reduced deformable objects. *ACM Trans. Graph. 32*, 6, Art. 213.

YUAN, Z., CHEN, F., AND ZHAO, Y. 2011. Pattern-guided smoke animation with Lagrangian Coherent Structure. *ACM Trans. Graph. 30*, 6, Art. 136.

YUDOVICH, V. 1963. Non-stationary flow of an ideal incompressible liquid. *USSR Computational Mathematics and Mathematical Physics 3*, 6, 1407–1456.

ZHANG, X., BRIDSON, R., AND GREIF, C. 2015. Restoring the missing vorticity in advection-projection fluid solvers. *ACM Trans. Graph. 34*, 4, Art. 52.

A Computing spectral bases

In this appendix, we describe how to compute the spectral bases for both vector fields and density fields on a mesh $\mathcal{M}$.

Discrete Laplacians. Finding our spectral bases first requires discretizing both the scalar Laplacian $\nabla\cdot\nabla$ and the vector Laplacian $-\nabla\times\nabla\times+\nabla\nabla\cdot$ on the domain $\mathcal{M}$. Discretization of these operators on arbitrary simplicial complexes is well documented [Desbrun et al. 2008; Elcott and Schröder 2006], and only involves topological operators d_1 and d_2 deriving from the mesh connectivity, and diagonal "Hodge star" operators $\star_1, \star_2$, and $\star_3$ based on local measures of $\mathcal{M}$ and its circumcentric dual, resulting in the following symmetric second-order operators:

$$\star_3\Delta_3 \equiv \star_3 d_2 \star_2^{-1} d_2^t \star_3, \quad \star_2\Delta_2 \equiv d_2^t \star_3 d_2 + \star_2 d_1 \star_1^{-1} d_1^t \star_2.$$

Note that d and $\star$ are even simpler on regular grids, even with the alteration we introduced in 3.4.

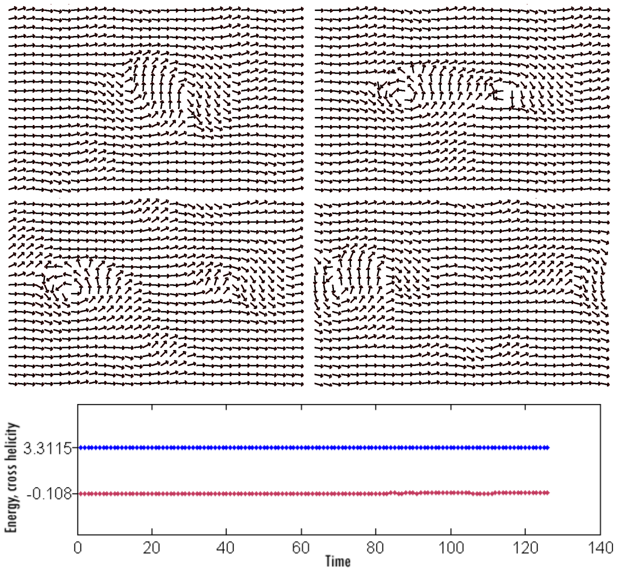


Figure 15: ***MHD rotor test****: The rotor test for magnetohydrodynamics consists of a dense rotating disk of fluid in an initially uniform magnetic field (left-right, top-middle: $t = 0.042, 0.126, 0.210, 0.336$). Our spectral integrator captures the correct behavior (see full dynamics in [Gawlik et al. 2011]) even with only 100 modes. Discrete energy (blue) and cross-helicity (red) are, as predicted, preserved over time (bottom).*

Boundary conditions. The canonical boundary conditions of velocity fields in fluid simulation for graphics purposes are no-transfer (i.e., the normal component v_n of the velocity along $\partial\mathcal{M}$ must be zero) and free-slip (i.e., the derivative of the tangential velocity field along the boundary normal $\partial v_t/\partial n$ must be zero as well). To enforce these conditions, we thus add the conditions that the flux of Ψ_i on *every boundary face* is zero, and that the circulations along the (interior half) boundary of the Voronoi face associated with *each boundary edge* is also zero (i.e, we simply set the values of $\star_1^{-1}$ as zeros for all the edges adjacent to the boundary faces to compute Δ_2). As for the eigenfunctions Φ_i of Δ_3, we use either Dirichlet boundary conditions $f|_{\partial M} = 0$ or Neumann boundary conditions $\frac{\partial f}{\partial n}|_{\partial M} = 0$, by considering boundary cell values or boundary gradients as null. If other, non-homogeneous boundary conditions (such as influx or outflux conditions) are required, then one must add an additional harmonic (zeroth frequency) component that satisfies the given boundary conditions.

Eigen computations. Once the Laplacians with proper boundary conditions are assembled, we can compute their low-frequency eigenfields using a simple Lanczos algorithm since these operators are symmetric. The constant eigenbases from the kernel of Δ_3 can be safely omitted by setting zero values on boundaries, since a constant scalar function is unchanged when advected by a divergence-free velocity field. Note that, as mentioned in §3.1, some of the eigenfields Ψ_i will be of the form $1/\mu_j\delta\Phi_j$: indeed, $\delta\Phi_i$ for $i \neq 0$ is an eigenfunction of Δ_2 since

$$\Delta_2\delta\Phi_j = \delta\Delta_3\Phi_j = \delta(-\mu_j^2\Phi_j) = -\mu_j^2\delta\Phi_j.$$

These gradient fields are easily identifiable by checking their divergence. Note finally that in theory, there could be cases where $\kappa_i^2 = \mu_j^2$ for multiple pairs of indices i and j. While in practice this is very unlikely to happen, one can protect against this rare event by replacing one of the corresponding Ψ_i by $\delta\Phi_j/\mu_j$, and replacing the other eigenvectors of this eigenvalue through a Gram-Schmidt process to form an orthonormal basis again.

Comments. We note that the approach we described above to compute the eigen bases for our fluid integrator is far from unique. For example, the β_1 harmonic vector fields are obtained as the eigenvectors associated with the eigenvalue 0, but we could have also computed the harmonic function dual to each homology generator via simple sparse linear systems instead [Tong et al. 2006]. Additionally, the vector field basis Ψ_i could be computed through its vector potential $\boldsymbol{\psi}_i$ instead: indeed, these vector potential 1-forms are eigenvectors of the 1-form Laplacian Δ_1, and boundary edge circulations as well as divergence on boundary dual cell divergence are assumed null to guarantee no-transfer and free-slip conditions. The curl of these 1-form basis functions are then, by construction, the flux-based Ψ_i basis functions. Finally, we point out that our approach is purposely different from what is proposed in [De Witt et al. 2012], as they use an eigendecomposition of $d\delta$ instead (leveraging the divergence-freeness of the vector fields). However, this simplified operator has a much larger null space that includes also curl-free fields, requiring many more eigenvectors to be computed via Lanczos iterations to generate divergence-free fields.

B Analysis of structural coefficients C_k

For simplicity, we use the periodic domain $[0,1]^3$, i.e., the flat 3D torus. The eigenfields can be expressed using complex numbers as

$$\Psi_i(\mathbf{x}) = \mathbf{w}_i e^{\jmath \mathbf{k}_i \cdot \mathbf{x}},$$

where $\jmath$ is the unit imaginary number, $\mathbf{x}$ is the 3D coordinates, $\mathbf{w}_i$ is a unit vector, and $\mathbf{k}_i$ is the wave number vector (i.e., with $|\mathbf{k}_i| = \kappa_i$). The divergence of the basis function is thus

$$\mathrm{div}\Psi_i(\mathbf{x}) = \jmath \mathbf{k}_i \cdot \mathbf{w}_i e^{\jmath \mathbf{k}_i \cdot \mathbf{x}},$$

while the curl is expressed as

$$\mathrm{curl}\Psi_i(\mathbf{x}) = \jmath \mathbf{k}_i \times \mathbf{w}_i e^{\jmath \mathbf{k}_i \cdot \mathbf{x}}.$$

Since $\mathrm{div}\Psi_i = 0$ means that $\mathbf{k}_i \cdot \mathbf{w}_i = 0$, there are two independent $\mathbf{w}$'s for each $\mathbf{k}$ in the basis. We can thus compute the structural coefficients $C_{c,ab}$ from Eq. (7) in closed form, by the integral of

$$(\nabla \times \Psi_a) \cdot (\Psi_c^* \times \Psi_b) = \jmath(\mathbf{k}_a \times \mathbf{w}_a) \cdot (\mathbf{w}_c \times \mathbf{w}_b) e^{\jmath(\mathbf{k}_a + \mathbf{k}_b - \mathbf{k}_c)\cdot \mathbf{x}},$$

where superscript * denotes complex conjugation. Note that nonzero coefficients $\jmath(\mathbf{k}_a \times \mathbf{w}_a) \cdot (\mathbf{w}_c \times \mathbf{w}_b)$ only exist when $\mathbf{k}_c = \mathbf{k}_a + \mathbf{k}_b$, and they advect real fields to real fields (whose coefficients satisfy $v_{\mathbf{k},\mathbf{w}} = v^*_{-\mathbf{k},\mathbf{w}}$). It indicates that $|\mathbf{k}_b|^2 C_{c,ab} = |\mathbf{k}_a|^2 C_{c,ba}$ is not true in general in 3D, contrary to the claim in [De Witt et al. 2012]; a simple counterexample is $\mathbf{k}_a = 2\pi(0,2,3)$, $\mathbf{w}_a = (1,0,0)$, $\mathbf{k}_b = 2\pi(1,1,0)$, and $\mathbf{w}_b = (0,0,1)$. Moreover, while $C_{c,aa} = 0$ indeed for this domain since $\mathbf{w}_c \cdot (2\mathbf{k}_a) = 0$, this property will no longer hold for an arbitrary domain. Thankfully, our variational integrator does not depend on the eigenmodes being steady flows, so these symmetries (or rather, lack thereof) are inconsequential.

C Kelvin's circulation theorem

Ideal, incompressible fluids have a conserved momentum [Chorin and Marsden 1979] given by the integrated circulation of the fluid around a closed curve which is advected by the flow. This fact is known as Kelvin's circulation theorem. Methods such as [Elcott et al. 2007] and [Pavlov et al. 2011] are constructed so as to conserve a discretized form of this conserved momentum. Our spectral method also obeys a form of Kelvin's theorem as follows. We can define generalized "spectral" curves as spectral dual 1-chains (also called 1-currents [Desbrun et al. 2008]) of the form:

$$\Gamma = \sum_i \gamma_i \star_2 A_i. \tag{10}$$

The above dual 1-chain expression always represents a closed curve, because each A_i corresponds to a closed (divergence-free) 2-form, which means the dual 1-chain is boundaryless. A pairing between a 2-form and a generalized loop is defined as expected:

$$\langle A, \Gamma \rangle = \left\langle \sum_i v_i A_i, \sum_j \gamma_j \star_2 A_j \right\rangle = \sum_i v_i \gamma_i. \tag{11}$$

The Lie advection of the generalized curve along the velocity field

$$\dot{\Gamma} = -[A, \Gamma] \tag{12}$$

indicates that the coefficients $\{\gamma_i\}_i$ must evolve such that

$$\begin{aligned} \dot{\gamma_k} &= -\sum_{i,j} \gamma_i v_j \int_{\mathcal{M}} \Psi_k \cdot (\nabla \times (\Psi_i \times \Psi_j)) \\ &= \sum_{i,j} \gamma_i v_j \int_{\mathcal{M}} (\nabla \times \Psi_k) \cdot (\Psi_j \times \Psi_i) = \sum_{i,j} \gamma_i v_j C_{j,ki}. \end{aligned}$$

Thus, the spectral version of Kelvin's theorem holds since

$$\begin{aligned} \frac{d}{dt}\langle A, \Gamma \rangle &= \sum_i (\dot{v_i}\gamma_i + v_i \dot{\gamma_i}) \\ &= \sum_i \mathbf{v}^t \mathbf{C}_i \mathbf{v} \gamma_i + \sum_i v_i \sum_{j,k} v_k \gamma_j C_{k,ij} \\ &= \sum_{i,j,k} v_j C_{i,jk} v_k \gamma_i - \sum_{i,j,k} v_j v_k \gamma_i C_{i,jk} = 0. \end{aligned}$$

In the above derivation, dummy index variables are swapped and the identity $C_{k,ij} = -C_{j,ik}$ is used.

D Temporal discretization

A fully discrete (in space and time) treatment of our variational integrator is easily achieved using the Hamilton-Pontryagin principle [Kharevych et al. 2006], where Lagrange multipliers μ^k enforce that A is indeed the Eulerian velocity of state q. If one denotes by h is the time step, A^k the velocity field between time k & state q^k and time $k+1$ & state q_{k+1}, the discrete Hamilton-Pontryagin action between $t = 0$ and $t = Nh$ is expressed as

$$S_d = \sum_{k=0}^{N-1} \frac{1}{2}\langle A^k, A^k \rangle h + \langle \mu^k, \tau^{-1}(q^{k+1}(q^k)^{-1}) - hA^k \rangle.$$

The map τ must convert an element of the Lie algebra to a Lie group element, thus making A_k the Eulerian velocity between time t_k and t_{k+1}; instead of the usual exponential map which is computationally difficult to handle, we approximate it to be the Cayley transform $\tau(A) = (I - A/2)^{-1}(I + A/2)$, as it efficiently maps antisymmetric matrices to orthogonal matrices. Taking variations with respect to μ^k, we recover the expected group element update rule

$$q^{k+1} = \tau(hA^k)q^k.$$

Variations with respect to A^k show that the multiplier is actually the momentum: $\mu^k = A^k$. Finally variations with respect to q^k restricted to $\delta q^k = B^k q^k$ with B^k in the Lie algebra (to enforce Lin constraints) yield

$$\begin{aligned} &\langle \mu^{k-1}, (I - hA^{k-1}/2)B^k(I + hA^{k-1}/2) \rangle \\ &\quad = \langle \mu^k, (I + hA^k/2)B^k(I - hA^k/2) \rangle. \end{aligned}$$

Omitting the cubic terms in $O(h^2)$ still preserves a discrete Kelvin's theorem, so we follow the suggestion in [Gawlik et al. 2011] and simplify the update rule to

$$\forall B^k, \langle A^{k-1}, B^k + \frac{h}{2}[B^k, A^{k-1}] \rangle = \langle A^k, B^k + \frac{h}{2}[B^k, -A^k] \rangle,$$

which reduces to the trapezoidal rule with $A^k = \sum_i v_i^k A_i$ and an arbitrary $B^k = \sum_i b_i^k A_i$.