

Accelerated Optimization in the PDE Framework Formulations for the Active Contour Case*

Anthony Yezzi[†], Ganesh Sundaramoorthi[‡], and Minas Benyamin[†]

Abstract. Following the seminal work of Nesterov, accelerated optimization methods have been used to powerfully boost the performance of first-order, gradient based parameter estimation in scenarios where second-order optimization strategies are either inapplicable or impractical. Not only does accelerated gradient descent converge considerably faster than traditional gradient descent, but it also performs a more robust local search of the parameter space by initially overshooting and then oscillating back as it settles into a final configuration, thereby selecting only local minimizers with a basis of attraction large enough to contain the initial overshoot. This behavior has made accelerated and stochastic gradient search methods particularly popular within the machine learning community. In their recent PNAS 2016 paper, *A Variational Perspective on Accelerated Methods in Optimization*, Wibisono, Wilson, and Jordan demonstrate how a broad class of accelerated schemes can be cast in a variational framework formulated around the Bregman divergence, leading to continuum limit ODEs. We show how their formulation may be further extended to infinite dimensional manifolds (starting here with the geometric space of curves and surfaces) by substituting the Bregman divergence with inner products on the tangent space and explicitly introducing a distributed mass model which evolves in conjunction with the object of interest during the optimization process. The coevolving mass model, which is introduced purely for the sake of endowing the optimization with helpful dynamics, also links the resulting class of accelerated PDE based optimization schemes to fluid dynamical formulations of optimal mass transport.

Key words. partial differential equations, acceleration, Nesterov, mass transport optimization, gradient descent, variational, manifolds

AMS subject classifications. 35B35, 49M99, 35J20, 35R30, 53C99, 65M99

DOI. 10.1137/19M1304210

1. Introduction. Following the seminal work of Nesterov, accelerated optimization methods (sometimes referred to as momentum methods) have been used to powerfully boost the performance of first-order, gradient based parameter estimation in scenarios where second-order optimization strategies are either inapplicable or impractical. Not only does accelerated gradient descent converge considerably faster than traditional gradient descent, but it also performs a more robust local search of the parameter space by initially overshooting and then oscillating back as it settles into a final configuration, thereby selecting only local minimizers with a basis of attraction large enough to contain the initial overshoot. This behavior

*Received by the editors December 6, 2019; accepted for publication (in revised form) July 8, 2020; published electronically November 19, 2020.

<https://doi.org/10.1137/19M1304210>

Funding: The work of the authors was supported by the Army Research Laboratory grant W911NF-18-1-0281 and the NIH grant R01-HL-143350.

[†]Department of Electrical and Computer Engineering, Georgia Institute of Technology 30332 USA (ayezzi@ece.gatech.edu, minasbenyamin@gmail.com).

[‡]Raytheon Technologies Research Center, East Hartford, CT 06118 USA (ganesh.sun@gmail.com).

has made accelerated and stochastic gradient search methods particularly popular within the machine learning community [31, 28, 27, 22, 21, 20, 16, 15, 6, 42]. So far, however, accelerated optimization methods have been restricted to searches over finite dimensional parameter spaces.

Recently, however, Wibisono, Wilson, and Jordan outlined a variational ODE framework in [59] (which we will summarize briefly in section 2.4) formulated around the Bregman divergence and which yields the continuum limit of a broad class of accelerated optimization schemes, including that of Nesterov's accelerated gradient method [32] whose continuum ODE limit was also demonstrated by Su, Boyd, and Candès in [48]. Here he will show how a similar high level framework may be adapted for infinite dimensional manifolds through the formulation of a generalized time-explicit action which can be viewed both as a specialization and generalization¹ of the Bregman Lagrangian presented in [59]. While the extension we outline from the ODE framework into the PDE framework is general enough to be applied to a variety of infinite dimensional or distributed-parameter optimization problems (dense shape reconstruction/inversion, optical flow estimation, image restoration, etc.), the specific examples presented here will focus on the active contour and active surface based optimization.²

mathematical, numerical, and computational challenges and technicalities which do not arise in finite dimensions. For example, the evolving parameter vector in finite dimensional optimization can naturally be interpreted as a single moving particle in \mathbf{R}^n with a constant mass which, in accelerated optimization schemes, gains momentum during its evolution. Since the mass is constant and fixed to a single particle, there is no need to explicitly model it. When evolving a continuous curve, surface, region, or function, however, the notion of accumulated momentum during the acceleration process is much more flexible, as the corresponding conceptual mass can be locally distributed in several different ways throughout the domain which will in turn significantly affect the evolution dynamics. In fact we intend to exploit this added design flexibility to further capture some of the same coarse-to-fine regularization properties of Sobolev gradient flows [60, 49] within the accelerated optimization context as well, but with far less computational cost.

The discrete implementation of accelerated PDE models will also differ greatly from existing momentum based gradient descent schemes in finite dimensions. Spatial and temporal steps sizes will be determined based on CFL stability conditions for finite difference approximations of the PDE's. Finally, in the PDE framework, viscosity solutions will be required in most cases to propagate through shocks and rarefactions that may occur during the evolution of a continuous front, a phenomenon which manifests itself differently and is therefore handled differently in the finite dimensional case. As such, these considerations will also impact the numerical discretization of accelerated PDE models.

Finally, in part due to these different discretization criteria and in part to avoid unnecessary complexity in the manifold case, we will abandon the Bregman Lagrangian described in [59] and will instead exploit a simpler time-explicit *generalized action* which will allow us to

¹We abandon the more general Bregman divergence in favor of simpler inner products, which, however, depend on the more general structure of the tangent space for the associated infinite dimensional manifold.

²A conference version of this paper was published in the IEEE Conference on Computer Vision and Pattern Recognition, 2019 [62].

work directly with the continuum velocity of the evolving entity rather than finite displacements with the Bregman divergence. Especially for the case of curves and surfaces considered here, this avoids the complication of calculating geodesic distances on highly curved, infinite-dimensional manifolds, but lets us work more easily in the tangent space instead.

2. Background and prior work. Geometric partial differential equations have played an important role in image analysis and computer vision for several decades now. Applications have ranged from low level processing operations such as denoising using anisotropic diffusion, blind deconvolution, and contrast enhancement; to midlevel processing such as segmentation using active contours and active surfaces, image registration, and motion estimation via optical flow; to higher level processing such as multiview stereo reconstruction, visual tracking, simultaneous localization and mapping (SLAM), and shape analysis. See, for example, [47, 46, 43] for introductions to PDE methods already established within computer vision within the 1990s, including level set methods [44] already developed in the 1980s for shape propagation. Several such PDE methods have been formulated, using the calculus of variations [57] as gradient descent based optimization problems in functional spaces, including geometric spaces of curves and surfaces.

During the past decade a popular trend has arisen whereby several such variational problems, which are nonconvex, have been reformulated and relaxed to convex optimization problems [10, 8, 18, 45], which allows one to build on the wealth of algorithms developed in the optimization literature [5]. While such methods have led to efficient and robust numerical schemes, the class of problems for which such reformulations apply is a limited class. We seek to develop optimization methods for a wider class of (nonconvex) problems.

Recently, Chaudhari et al. have established connections between relaxation techniques used in training deep neural networks and PDE's in [13] based on the continuum Fokker–Planck equation limit. They, in turn, develop and demonstrate improved implementations of stochastic gradient descent based on the viscous Hamilton–Jacobi equation. Subsequently, in [14], Chaudhari and Soatto demonstrate that stochastic gradient descent (SGD) methods perform variational inference (although not on the original loss function). While they do exploit momentum to accelerate convergence in their numerical algorithms, this acceleration component is introduced on the backend of the final discrete algorithm. The methodology presented in section 3.2.3, through the incorporation of an auxiliary evolving density function, offers a potential strategy to directly integrate acceleration into their original continuum PDE formulation of SGD as well. However, our focus here will remain exclusively on acceleration, by itself, within the continuum PDE framework.

2.1. Geometric active contours (an example of gradient PDE optimization). For example, several active contour models are formulated as gradient descent PDE flows of application-specific energy functionals E which relate the unknown contour C to given data measurements. Such energy functionals are chosen to depend only upon the geometric shape of the contour C , not its parameterization. Under these assumptions the first variation of E will have the following form:

$$(2.1) \quad \delta E = - \int_C f(\delta C \cdot N) ds,$$

where fN represents a perturbation field along the unit normal N at each contour point

and ds denotes the arc length measure. Note that the first variation depends only upon the normal component of a permissible contour perturbation δC . The form of f will depend upon the particular choice of the energy. For example, in the popular Chan–Vese active contour model [9] for image segmentation, f would be expressed by $(I - c_1)^2 - (I - c_2)^2 + \alpha\kappa$, where I denotes the image value at a given contour point, α an arc length penalty weight, κ the curvature at a given contour point, and c_1 and c_2 the means of the image inside and outside the contour, respectively. As an alternative example, the geodesic active contour model [7, 25] would correspond to $f = \phi\kappa N - (\nabla\phi \cdot N)N$, where $\phi > 0$ represents a point measurement designed to be small near a boundary of interest and large otherwise. In all cases, though, the gradient descent PDE will have the following explicit form:

$$(2.2) \quad \frac{\partial C}{\partial t} = fN \quad (\text{explicit gradient flow}).$$

This class of contour flows, evolving purely in the normal direction, may be implemented implicitly in the level set framework [44] by evolving a function ψ whose zero level set represents the curve C as follows:

$$\frac{\partial \psi}{\partial t} = -\hat{f}\|\nabla\psi\| \quad (\text{implicit level set flow}),$$

where $\hat{f}(x, t)$ denotes a spatial extension of $f(s, t)$ to points away from the curve.

2.2. Sobolev gradients for more robust coarse-to-fine PDE based optimization. The most notorious problem with most active contour and active surface models is that the normal speed function f depends pointwise upon noisy or irregular data measurements, causing immediate fine scale perturbations in the evolving contour which cause it to become very easily attracted to (and trapped within) spurious local minimizers. This often makes the active contour model strongly dependent upon initialization, except for a limited class of convex or polyconvex energy functionals for which numerical schemes can be devised to reach global minimizers reliably. The traditional way to combat this sensitivity is to add strong regularizing terms to the energy functional which penalize fine scale irregularities in the contour shape. Similar problems and regularization strategies are applied in other PDE based optimization applications outside the realm of the illustrative active contour example being considered here (for example, in Horn and Schunck style optical flow computation [19]).

This energy regularization strategy has two drawbacks. First, most regularizers lead to second order (or higher) diffusion terms in the gradient contour flow, which impose much smaller time step limitations on the numerical discretization of the evolution PDE. Thus, significantly more evolution steps are required, which incurs a heavy computational cost in the minimization process. Second, regularizers, while endowing a level of resistance to noise and spurious structure, impose regularity on the final converged contour as well, making it difficult or impossible to capture features such as sharp corners or narrow protrusions/inlets in the detected shape. This can lead to unpleasant trade-offs in several applications.

For the illustrative case of active contours, significantly improved robustness in the gradient flow, without additional energy regularization, can be attained by using geometric Sobolev gradients [12, 11, 53, 54] in place of the standard L^2 -style gradient used in traditional active

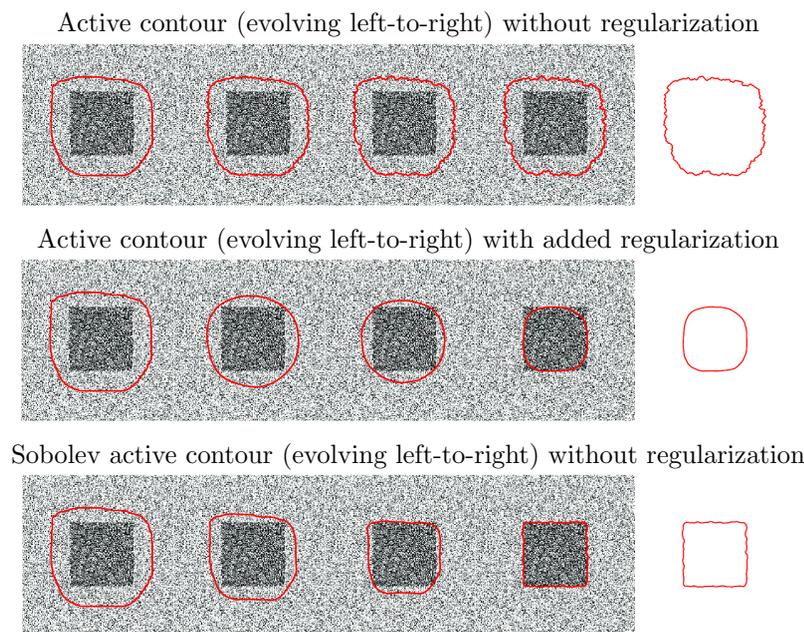


Figure 1. Sobolev gradients versus energy regularization.

contours. We refer to this class of active contours as Sobolev active contours, whose evolution may be described by the following integral PDE

$$(2.3) \quad \frac{\partial C}{\partial t} = (fN) * K \quad (\text{Sobolev gradient flow}).$$

Here $*$ denotes convolution in the arc length measure with a smoothing kernel K to invert the linear Sobolev gradient operator. The numerical implementation is not carried out this way, but the expression gives helpful insight into how the Sobolev gradient flow (2.3) relates to the usual gradient flow (2.2). Namely, the optimization process (rather than the energy functional itself) is regularized by averaging pointwise gradient forces fN through the kernel K to yield a smoother contour evolution. This does not change the local minimizers of the energy functional, nor does it impose extra regularity at convergence, but it induces a coarse-to-fine evolution behavior [49, 55, 60] in the contour evolution, making it much more resistant to spurious local minima due to noise or other fluctuations in f .

The regularity of the coarse-to-fine Sobolev gradient flow compared with regularity imposed on the energy functional is illustrated in Figure 1. Along the top row we see the evolution of a standard active contour in a very noisy image without regularizing terms in the energy function to keep the contour smooth. The contour quickly gets trapped in a noisy local minimum configuration before reaching the desired square boundary. Of course, we can add a regularizing term to the energy to prefer smoother contours. We see in the middle row that this fixes the noise problem but does not allow us to capture the sharp corners of the square. Along the bottom row, instead, we show the evolution of the Sobolev active contour for the original *unregularized* energy from the top row. The initial stages of the evolution maintain a smooth contour, not because the Sobolev gradient prefers a smooth contour, but because it

prefers a smooth evolution. As the Sobolev active contour nears the boundary of the square, finer scale motions are incorporated to bring out the corners. The final converged contour responds to local noise, but only in the vicinity of a desired minimizer.

However, while the Sobolev gradient descent method is extremely successful in making an active contour or surface (or other evolving classes of functions) resistant to a large class of unwanted local minimizers, it comes at a heavy computational cost. The spatial integration of gradient forces along the evolving front must occur during every time step, and while there are tricks to do this quickly for closed 2 dimensional (2D) curves [29, 51, 50, 2] there are no convenient alternatives for 3 dimensional (3D) surfaces, nor for regions (even in 2 dimensions) when applying Sobolev gradient flows to other functional objects (images, optical flow, etc.). The linear operator inversion imposes a notable per-iteration cost, which we will instead distribute across iterations in the upcoming accelerated coupled PDE evolution schemes.

2.3. Momentum methods and Nesterov's accelerated gradient. If we step back to the finite dimensional case, an alternative and computationally cheaper method to regularize any gradient descent based iteration scheme is to employ the use of momentum. In such schemes the new update becomes a weighted combination of the previous update (the momentum term) and the newly computed gradient at each step. This leads to a temporal averaging of gradient information computed and accumulated during the evolution process itself, rather than a spatial averaging that occurs independently during each time step. As such it adds an insignificant per iteration computation cost while significantly boosting the robustness (and often the convergence speed) of the optimization process.

Momentum methods, including stochastic variants [16, 20], have become very popular in machine learning in recent years [6, 15, 21, 22, 27, 31, 42, 28]. Strategic dynamically changing weights on the momentum term can further boost the descent rate. Nesterov put forth the following famous scheme [32] which attains an optimal rate of order $\frac{1}{t^2}$ in the case of a smooth, convex energy function $E(x)$:

$$y_{k+1} = x_k - \frac{1}{\beta} \nabla E(x_k), \quad x_{k+1} = (1 - \gamma_k) y_{k+1} + \gamma_k y_k,$$

$$\gamma_k = \frac{1 - \lambda_k}{\lambda_k + 1}, \quad \lambda_k = \frac{1 + \sqrt{1 + 4\lambda_{k-1}^2}}{2},$$

where x_k is the k th iterate of the algorithm, y_k is an intermediate sequence, and γ_k are dynamically updated weights.

2.4. A variational framework for accelerated ODE optimization. Recently in [59] Wibisono, Wilson, and Jordan presented a variational generalization of Nesterov's [32] and other momentum based gradient descent schemes in \mathbb{R}^n based on the Bregman divergence of a convex distance generating function h ,

$$(2.4) \quad D(y, x) = h(y) - h(x) - \langle \nabla h(x), y - x \rangle,$$

and careful discretizations of the Euler–Lagrange equation for the time integral (evolution time) of the following Bregman Lagrangian,

$$\mathcal{L}(X, V, t) = e^{a(t)+\gamma(t)} \left[D(X + e^{-a(t)}V, X) - e^{b(t)}\mathbf{U}(X) \right],$$

where the potential energy \mathbf{U} represents the cost to be minimized. In the Euclidean case, where $D(y, x) = \frac{1}{2}\|y - x\|^2$, this simplifies to

$$\mathcal{L} = e^{\gamma(t)} \left[e^{-a(t)} \underbrace{\frac{1}{2}\|V\|^2}_{\mathbf{T}} - e^{a(t)+b(t)} \mathbf{U}(X) \right],$$

where \mathbf{T} models the kinetic energy of a unit mass particle in \mathbb{R}^n . Nesterov's methods [32, 36, 35, 34, 37, 33] belong to a subfamily of Bregman Lagrangians with the following choice of parameters (indexed by $k > 0$),

$$a = \log k - \log t, \quad b = k \log t + \log \lambda, \quad \gamma = k \log t,$$

which, in the Euclidean case, yields a time-explicit generalized action (compared to the time-implicit standard action $\mathbf{T} - \mathbf{U}$ from classical mechanics [17]) as follows:

$$(2.5) \quad \mathcal{L} = \frac{t^{k+1}}{k} \left(\mathbf{T} - \lambda k^2 t^{k-2} \mathbf{U} \right).$$

In the case of $k = 2$, for example, the Euler–Lagrange equations for the integral of this time-explicit action yield the continuum limit of Nesterov's accelerated mirror descent [33] derived in both [48, 27]. The PDE acceleration framework that we present here is also extended to the case of diffeomorphisms [64, 56, 52] in concurrent work, and linear function spaces in [4].

3. Accelerated optimization in the PDE framework. We now develop a general strategy, based on a generalization of the Euclidean case of Wibisono, Wilson, and Jordan's formulation [59] reviewed in section 2.4, for extending accelerated optimization into the PDE framework. While our approach will be motivated by the variational ODE framework formulated around the Bregman divergence in [59], we will have to address several mathematical, numerical, and computational considerations which do not need to be addressed in finite dimensions.

For example, the evolving parameter vector in finite dimensional optimization can naturally be interpreted as a single moving particle in \mathbf{R}^n with a constant mass which, in accelerated optimization schemes, gains momentum during its evolution. Since the mass is constant and fixed to a single particle, there is no need to explicitly model it. When evolving a continuous curve, surface, region, or function, however, the notion of accumulated momentum during the acceleration process is much more flexible, as the corresponding conceptual mass can be locally distributed in several different ways throughout the domain which will in turn significantly affect the evolution dynamics. We outline two different mass models in sections 3.2.2 and 3.2.3 as starting points and show how additional control of the optimization dynamics can be introduced in conjunction with the more flexible second mass model by considering independent mass-related potential energy terms in section 3.5.2. In all cases, the outcome of these formulations will be a coupled system of first-order PDEs which govern the simultaneous evolution of the continuous unknown (curves in the case considered here), its velocity, as well as the supplementary density function which describes the evolving mass.

In addition, as pointed out from the onset, the numerical discretization of accelerated PDE models will also differ greatly from existing momentum based gradient descent schemes

in finite dimensions. Spatial and temporal steps sizes will be determined based on CFL stability conditions for finite difference approximations of the PDEs and viscosity solution schemes will be required in most cases to propagate through shocks and rarefactions that may occur during the evolution of a continuous front. This is part of the reason we replace the more general Bregman–Lagrangian in [59] with the simpler time-explicit generalized action (2.5), together with the additional benefit that such a choice allows us to work directly with the continuum velocity of the evolving entity (or other generalizations that are easily defined within the tangent space of its relevant manifold) rather than finite displacements utilized by the Bregman divergence (2.4).

3.1. General approach. Just as in [59], the energy functional E to be optimized over the continuous infinite dimensional unknown (whether it be a function, a curve, a surface, or a diffeomorphic mapping) will represent the potential energy term \mathbf{U} in the time-explicit generalized action (2.5). Next, a customized kinetic energy term \mathbf{T} will be formulated to incorporate the dynamics of the evolving estimate during the minimization process. Note that just as the evolution time t would represent an artificial time parameter for a continuous gradient descent process, the kinetic energy term will be linked to artificial dynamics incorporated into the accelerated optimization process. As such, the accelerated optimization dynamics can be designed completely independently of any potential physical dynamics in cases where the unknown might be connected with the motion of real objects. Several different strategies can be explored, depending upon the geometry of the specific optimization problem, for defining kinetic energy terms, including various approaches for attributing artificial mass (both its distribution and its flow) to the actual unknown of interest in order to boost the robustness and speed of the optimization process.

Once the kinetic energy term has been formulated, the accelerated evolution will be obtained (prior to discretization) using the calculus of variations [57] as the Euler–Lagrange equation of the following time-explicit *generalized action integral*

$$(3.1) \quad \int \frac{t^{k+1}}{k} \left(\mathbf{T} - \lambda k^2 t^{k-2} \mathbf{U} \right) dt.$$

In the simple $k = 2$ case, the main difference between the resulting evolution equations versus the classical principle of least action equations of motion (without the time explicit terms in the Lagrangian) is an additional friction-style term whose coefficient of friction decreases inversely proportional to time. This additional term, however, is crucial to the accelerated minimization scheme. Without such a frictional term, the Hamiltonian of the system (the total energy $\mathbf{T} + \mathbf{U}$), would be conserved, and the associated dynamical evolution would never converge to a stationary point. Friction guarantees a monotonic dissipation of energy, allowing the evolution to converge to a state of zero kinetic energy and locally minimal potential energy (the optimization objective).

This yields a natural physical interpretation of accelerated gradient optimization in terms of a mass rolling down a potentially complicated terrain by the pull of gravity (Figure 2). In gradient descent, its mass is irrelevant, and the ball always rolls downward by gravity (the gradient). As such the gradient directly regulates its velocity. In the accelerated case, gravity

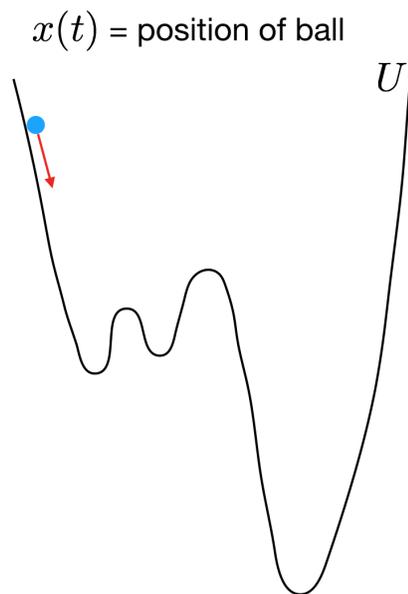


Figure 2. Accelerated descent physics interpretation.

regulates its acceleration. Friction can be used to interpolate these behaviors, with gradient descent representing the infinite frictional limit as pointed out in [59].

Acceleration comes with two advantages. First, whenever the gradient is very shallow (the energy functional is nearly flat), acceleration allows the ball to accumulate velocity as it moves so long as the gradient direction is self-reinforcing. As such, the ball approaches a minimum more quickly. Second, the velocity cannot abruptly change near a shallow minimum as in gradient descent. Its mass gives it momentum, and even if the acceleration direction switches in the vicinity of a shallow minimum, the accumulated momentum still moves it forward for a certain amount of time, allowing the optimization process to *look ahead* for a potentially deeper minimizer.

3.2. Accelerated active contours. We now illustrate the steps in the process for developing PDE based accelerated optimization schemes for the specific case of geometric active contours. The resulting coupled PDE evolutions will retain the parameterization independent property of gradient descent based active contours models and will therefore remain amenable to implicit implementation using level set methods [44].

We begin, however, by reviewing some basic differential contour evolution properties that will be useful in deriving accelerated active contour formulations. In particular, it is useful to understand any contour evolution behavior in terms of its local geometric frame, consisting of the unit tangent and normal vectors.

Let $C(p, t)$ denote an evolving curve where t represents the evolution parameter and $p \in [0, 1]$ denotes an independent parameter along each fixed curve. The unit tangent, unit normal, and curvature will be denoted by $T = \frac{\partial C}{\partial s}$, N , and κ , respectively, with the sign convention for κ and the direction convention for N chosen to respect the planar Frenet

equations $\frac{\partial T}{\partial s} = \kappa N$ and $\frac{\partial N}{\partial s} = -\kappa T$, where s denotes the time-dependent arc length parameter whose derivative with respect to p yields the parameterization speed $\frac{\partial s}{\partial p} = \|\frac{\partial C}{\partial p}\|$.

Letting α and β denote the tangential and normal speeds of the evolving curve,³

$$(3.2) \quad \frac{\partial C}{\partial t} = \alpha T + \beta N,$$

the frame itself can be shown to evolve as follows:

$$(3.3) \quad \frac{\partial T}{\partial t} = \left(\frac{\partial \beta}{\partial s} + \alpha \kappa \right) N, \quad \frac{\partial N}{\partial t} = - \left(\frac{\partial \beta}{\partial s} + \alpha \kappa \right) T.$$

Differentiating the velocity decomposition (3.2) with respect to t , followed by the frame evolution (3.3) substitution, yields the acceleration

$$(3.4) \quad \frac{\partial^2 C}{\partial t^2} = \left(\frac{\partial \alpha}{\partial t} - \beta \left(\frac{\partial \beta}{\partial s} + \alpha \kappa \right) \right) T + \left(\frac{\partial \beta}{\partial t} + \alpha \left(\frac{\partial \beta}{\partial s} + \alpha \kappa \right) \right) N$$

which may be rewritten as the following two scalar evolution equations for the tangential and normal speeds, in terms of the tangential and normal components of the contour acceleration, respectively:

$$(3.5) \quad \frac{\partial \alpha}{\partial t} = \frac{\partial^2 C}{\partial t^2} \cdot T + \beta \left(\frac{\partial \beta}{\partial s} + \alpha \kappa \right), \quad \frac{\partial \beta}{\partial t} = \frac{\partial^2 C}{\partial t^2} \cdot N - \alpha \left(\frac{\partial \beta}{\partial s} + \alpha \kappa \right).$$

3.2.1. Contour potential energy. For geometric active contours, we start by defining the potential energy \mathbf{U} to be an originally provided energy functional E which depends only upon the geometric shape of the contour C (not its parameterization). Under these assumptions the first variation of the potential energy will have the following form, just as in (2.1) presented earlier in section 2.1, where fN denotes the backward local gradient force at each contour point:

$$\delta \mathbf{U} = - \int_C f (\delta C \cdot N) ds$$

3.2.2. Constant density model. To formulate an accelerated evolution model, we define a kinetic energy, which requires a notion of mass coupled with velocity. The simplest starting model would be one of constant mass density ρ (per unit arc length along the contour) and an integral of the squared norm of the pointwise contour evolution velocity:⁴

$$(3.6) \quad \mathbf{T} = \frac{1}{2} \rho \int_C \left(\frac{\partial C}{\partial t} \cdot \frac{\partial C}{\partial t} \right) ds.$$

³Note that the instantaneous geometric deformation of the curve is determined exclusively by the normal speed β , and that gradient flows for geometric active contours can all be formulated such that the tangential speed α vanishes. We will see later that the same is possible for accelerated flow models as well.

⁴A similar kinetic energy model in the context of the classical action $\mathbf{T} - \mathbf{U}$, for example, was used to develop dynamic geodesic snake models for visual tracking in [40].

Plugging this into the generalized action integral (3.1) and computing the Euler–Lagrange equation leads to our first, and simplest, accelerated model:

$$(3.7) \quad \underbrace{\frac{\partial^2 C}{\partial t^2}}_{\text{acceleration}} = \frac{\lambda k^2 t^{k-2}}{\rho} \underbrace{fN}_{-\text{gradient}} - \underbrace{\left(\frac{\partial^2 C}{\partial s \partial t} \cdot \frac{\partial C}{\partial s} \right) \frac{\partial C}{\partial t} - \frac{\partial}{\partial s} \left(\frac{1}{2} \left\| \frac{\partial C}{\partial t} \right\|^2 \frac{\partial C}{\partial s} \right)}_{\text{wave propagation terms}} - \underbrace{\frac{k+1}{t} \frac{\partial C}{\partial t}}_{\text{friction}}.$$

If we start with zero initial velocity we can decompose this nonlinear second-order PDE into the following coupled system of nonlinear first-order PDEs,

$$(3.8) \quad \frac{\partial C}{\partial t} = \beta N, \quad \frac{\partial \beta}{\partial t} = \frac{\lambda k^2 t^{k-2}}{\rho} f + \frac{1}{2} \beta^2 \kappa - \frac{k+1}{t} \beta.$$

Since the contour evolution remains purely geometric (only in the normal direction N) we may also write down an implicit level set version of the coupled PDE system as follows:

$$(3.9) \quad \frac{\partial \psi}{\partial t} = \hat{\beta} \|\nabla \psi\|, \quad \frac{\partial \hat{\beta}}{\partial t} = \frac{\lambda k^2 t^{(k-2)}}{\rho} \hat{f} + \nabla \cdot \left(\frac{1}{2} \hat{\beta}^2 \frac{\nabla \psi}{\|\nabla \psi\|} \right) - \frac{k+1}{t} \hat{\beta},$$

where $\hat{f}(x, t)$ and $\hat{\beta}(x, t)$ denote spatial extensions of f and β , respectively.

3.2.3. Conserved flowable mass model. The kinetic energy in the accelerated formulation is invented to endow the minimization evolution process with helpful dynamics for the sake of faster and more robust convergence. Thus, just as the potential energy does not actually represent a real physical energy, there is no need to impose real physical considerations on the kinetic energy either. Nonetheless, the simple constant density model feels quite unnatural in that it does not preserve total mass if the contour length changes during its evolution: mass is created when the contour expands and is destroyed when the contour contracts.

A more flexible and natural way to attribute mass to the evolving contour is to consider an arbitrary and independent distribution of mass along the contour which evolves as the curve evolves. As such, the mass density ρ can vary both spatially and temporarily, while the total integrated mass is still conserved. In such a model, though, not only does mass evolve as a result of contour shape deformation, but it may also flow along the contour without changing its geometry (therefore contributing to the kinetic energy without affecting the potential energy). A simple interpretation would be that the contour shape represents a moving container for a fluid which not only gets pushed around by the extrinsic motion of the container but which may also flow with an independent relative internal speed v inside of the container (i.e., along the tangent direction of the contour). As such, the velocity of each mass particle at a given contour point would be the sum of the contour velocity and the internal mass flow velocity:

$$v = \text{internal mass flow speed}, \quad \text{total mass velocity} = \frac{\partial C}{\partial t} + v \frac{\partial C}{\partial s}.$$

This suggests a more general kinetic energy model as follows:

$$(3.10) \quad \mathbf{T} = \int_C \frac{1}{2} \rho \left\| \frac{\partial C}{\partial t} + v \frac{\partial C}{\partial s} \right\|^2 ds$$

but with the density evolution constrained by the following continuity equation to ensure local conservation of mass:

$$(3.11) \quad \underbrace{\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial s}(\rho v)}_{\text{mass change}} + \rho \underbrace{\left(\frac{\partial^2 C}{\partial s \partial t} \cdot \frac{\partial C}{\partial s} \right)}_{\text{length change}} = 0.$$

The latter may be incorporated as a Lagrange multiplier constraint when computing the Euler–Lagrange equation of the generalized action integral (3.1). This results in the following second-order PDEs which, together with (3.11), yield the accelerated system as a coupled evolution of C along with the auxiliary mass density ρ and internal flow v field responsible for these helpful dynamics:

$$(3.12) \quad \begin{aligned} \frac{\partial^2 C}{\partial t^2} \cdot N &= - \left(2v \frac{\partial^2 C}{\partial s \partial t} + v^2 \frac{\partial^2 C}{\partial s^2} + \frac{k+1}{t} \frac{\partial C}{\partial t} \right) \cdot N + \frac{\lambda k^2 t^{k-2} f}{\rho}, \\ \frac{\partial v}{\partial t} &= - \left(\frac{\partial^2 C}{\partial t^2} + v \frac{\partial^2 C}{\partial s \partial t} + \frac{k+1}{t} \frac{\partial C}{\partial t} \right) \cdot \frac{\partial C}{\partial s} - \left(\frac{\partial v}{\partial s} + \frac{k+1}{t} \right) v. \end{aligned}$$

Notice in this flowable conserved mass model, that only the normal component of the curve acceleration $\frac{\partial^2 C}{\partial t^2}$ is governed by the Euler–Lagrange equation. The tangential acceleration, even though it affects the internal mass flow, can be chosen freely. We may exploit this degree of freedom to keep the tangential velocity of the curve equal to zero, thus keeping the evolution purely geometric. Accordingly, and just as in the constant density case, we may convert the second-order system (3.12) into a first-order geometric system of PDEs. In particular, if we start out with zero initial velocity, we obtain the following equivalent system of three coupled evolution PDEs for C , V , and ρ which, in contrast with the constant density scheme, also avoids the calculation of curvature:

$$(3.13) \quad \begin{aligned} \underbrace{\frac{\partial V}{\partial t}}_{\text{acceleration}} &= \frac{\lambda k^2 t^{k-2}}{\rho} \underbrace{\hat{f} N}_{\text{-gradient}} - \underbrace{\left(V \cdot \frac{\partial C}{\partial s} \right) \frac{\partial V}{\partial s}}_{\text{advection}} - \underbrace{\frac{k+1}{t} V}_{\text{friction}}, \\ \frac{\partial C}{\partial t} &= \underbrace{(V \cdot N)}_{\beta} N, \quad \frac{\partial \rho}{\partial t} = - \underbrace{\left(V \cdot \frac{\partial C}{\partial s} \right) \frac{\partial \rho}{\partial s} - \rho \frac{\partial V}{\partial s} \cdot \frac{\partial C}{\partial s}}_{\text{mass preservation}}. \end{aligned}$$

Here the velocity field defined as $V = vT + \beta N$ captures both the tangential flow of the mass as well as the normal flow of the curve itself. As in the constant density model, we see that the evolution of the contour remains purely geometric (only in the normal direction), and thus with suitable spatial extension functions $\hat{V}(x, t)$ and $\hat{\rho}(x, t)$ this system can easily be adapted to the level set framework as well. One notable difference, however, is that the evolution equation for the level set function ψ itself, becomes linear in this case:

$$(3.14) \quad \frac{\partial \psi}{\partial t} = \hat{V} \cdot \nabla \psi.$$

3.3. Mixing optimization dynamics with physical time dynamics. Again, while the kinetic energy models, including their attributed mass density functions ρ , are invented purely for the sake of improved optimization, there may be applications in which physically meaningful considerations could nonetheless be usefully blended into the optimization dynamics. Two particular application areas where very strong connections could be made include dynamic tracking as well as optimal mass transport.

3.3.1. Connections with dynamic tracking. Niethammer and Tannenbaum [38, 39, 40] introduced a new geometric dynamical active contour model that has strong connections to the present work. The motivation is visual tracking. The authors make the point that the use of active contours is typically preformed statically. More specifically, the active contour captures the given object at a certain time t and then some prediction procedure is employed to give a reasonable initial placement at time $t + 1$. The problem is that the curve evolution gets decoupled from the dynamics of the target. The standard dynamic approaches are marker particle based and thus lose the advantages of the level set methodology, the shortcomings of such particle-based implementations. The works of [38, 39, 40] develop a straightforward, efficient, level set based approach for dynamic curve evolution which removes the separation of segmentation and prediction, while preserving the many advantages of level set formulations. The key idea is based on the minimization of a novel energy functional that adds dynamics into the geodesic active contour framework.

More precisely, the above approach develops dynamical geodesic snake models for visual tracking based on the classical action $\mathbf{T} - \mathbf{U}$ using constant density mass models. This endowed the moving contour with dynamics in actual *physical time* which could be used in the context of dynamic observers [41].

Such a scheme for frame-to-frame evolution of a contour within a video would pair very naturally, for example, with the simplest-case optimization dynamics from section 3.2.2 using the same kinetic energy model (3.6), but in the context of the generalized action (2.5) for static optimization within each individual video frame. Conversely, the more general kinetic energy models outlined in section 3.2.3 for optimization using the generalized action, could be similarly be adapted to the the problem of visual tracking using the classical action.

3.3.2. Connections with optimal mass transport. The conserved evolutionary mass model underlying the accelerated system (3.13) begins to exhibit clear connections to problems in optimal mass transport [24, 3, 1, 58], especially in the fluid-dynamical formulation of Benamou and Brenier [3].

Optimal mass transport is a very old problem first introduced by the civil engineer Monge in 1781 [30] and concerned with finding the optimal way, in the sense of minimal transportation cost, of moving a pile of soil from one site to another. This problem of *optimal mass transport* was given a modern formulation in the work of Kantorovich [23, 24], and so is now known as the Monge–Kantorovich problem. As originally formulated, the problem has no explicit dynamics, and basically leads to a metric on probability densities, the *Wasserstein distance*. Optimal mass transport is a very active area of research with application to numerous disciplines including probability, econometrics, fluid dynamics, automatic control, transportation, statistical physics, shape optimization, expert systems, and meteorology.

A major development in optimal mass transport theory was realized in the seminal dynamic approach to optimal mass transport by Benamou and Brenier [3]. These authors base their approach to optimal mass transport on ideas from fluid mechanics via the minimization of a kinetic energy functional subject to a continuity constraint.

The work described above is very much in line with the latter dynamics approach. In fact, given that the mass is introduced as an independent auxiliary variable for the sake of acceleration, we may just as easily allow it to live within the contour interior rather than along the contour boundary. The resulting region based extension of the kinetic energy model (3.10) would then match the functional whose minimizer, as demonstrated Benamou and Brenier, yields a flow of diffeomorphisms which minimize the Wasserstein distance between the mass distributions at any two instances along its trajectory (including the initial and final distributions).

3.4. Accelerated active surfaces. The accelerated active contour models developed in section 3.2 offer a more robust evolution framework for generic contour based optimization problems, just as the class of Sobolev active contour models introduced earlier. Both methodologies regularize the optimization process, without imposing regularity on the final optimized result, greatly boosting the evolving contour's resistance to spurious or shallow local minimizers. In both cases, this desirable property is achieved by effectively averaging contributions from several local gradient forces in order to determine the instantaneous evolution of any given point on the curve.

In the case of Sobolev active contours, this averaging is done spatially at each fixed time instant by an effective convolution along the curve. Unfortunately, while special tricks exist to do this quickly for closed curves, they do not apply to surfaces or higher dimensional manifolds, where Laplace–Beltrami style PDEs must instead be solved along the surface at every time instant in order to calculate the Sobolev gradient.

Accelerated active contour models, on the other hand, perform a temporal rather than spatial averaging. As a particle along the curve accelerates, its instantaneous velocity represents the accumulation of local gradient information over its recently traveled trajectory, rather than the accumulation of local gradient information from its neighboring contour points at the same instant in time. An important advantage of the time based averaging, in contrast to the instantaneous spatially based averaging⁵ in Sobolev style active contours, is that the same computational speed up in 2 dimensions will apply equally in 3 dimensions and higher.

In the case of geometric active surfaces, we start with a potential energy which depends only upon the geometric shape of the contour S (again, as in the contour case, not its parameterization). Under these assumptions the first variation of the potential energy will have the following form,

$$\delta\mathbf{U} = - \int_S f(\delta C \cdot N) dA,$$

where fN represents a force along the unit normal N at each point on the surface S and where dA denotes the surface area measure. The implicit level set framework is particularly

⁵In section 3.5 we show how additional strategies within the accelerated framework can be devised to further incorporate some level of spatial averaging, thereby obtain the maximum amount of evolution robustness and leveraging the best of both Sobolev and accelerated optimization yet without the added computational cost of inverting the Sobolev operator.

convenient for active surfaces given the complexities of dealing with 3D meshes. In the level set framework, the (nonaccelerated) gradient descent surface evolution PDE has the same form as in 2 dimensions, but is applied to a 3D grid instead. Namely,

$$\frac{\partial \psi}{\partial t} = -\hat{f} \|\nabla \psi\|,$$

where $\hat{f}(x, t)$ denotes a spatial extension of f to points away from the surface. Narrow band methods are especially important in 3 dimensions to keep the computational cost of updating the level set function ψ to a minimum (as well as limiting the neighborhood where extension functions such as \hat{f} need to be computed and evolved).

In the simplest constant density model case, applied to surfaces, the kinetic energy term for the accelerated model will have a similar form but with the density ρ interpreted per unit surface area as

$$\mathbf{T} = \frac{1}{2} \rho \int_S \left(\frac{\partial S}{\partial t} \cdot \frac{\partial S}{\partial t} \right) dA.$$

Computing the Euler–Lagrange equation of the generalized action integral (3.1) and writing it in the level set framework yields the same system of first-order PDEs as in the contour case, except now in 3 dimensions,

$$\frac{\partial \psi}{\partial t} = -\hat{\beta} \|\nabla \psi\|, \quad \frac{\partial \hat{\beta}}{\partial t} = \frac{\lambda k^2 t^{(k-2)}}{\rho} \hat{f} + \nabla \cdot \left(\frac{1}{2} \hat{\beta}^2 \frac{\nabla \psi}{\|\nabla \psi\|} \right) - \frac{k+1}{t} \hat{\beta},$$

where $\hat{f}(x, t)$ and $\hat{\beta}(x, t)$ denote 3D spatial extensions of f and β , respectively.

3.5. Acceleration with spatial regularity (capturing Sobolev gradient properties). There are several ways in the PDE framework that we may seek to combine the spatial averaging of gradient information inherent to Sobolev gradient descent with the temporal averaging of gradient information inherent to acceleration, while still remaining fully within the accelerated framework, bypassing the linear operator inversion required in the Sobolev framework. We present two different strategies for obtaining the best of both.

3.5.1. Adding velocity diffusion. A simple way to incorporate spatial averaging in the acceleration process would be to heuristically add a diffusion term in the velocity update. For a concrete example, in the conserved flowable-mass acceleration strategy for active contours outlined in section 3.2.3, we could augment the acceleration PDE (3.13) as follows (the coupled density evolution PDE would remain the same):

$$\overbrace{\frac{\partial V}{\partial t}}^{\text{acceleration}} = \frac{\lambda k^2 t^{k-2}}{\rho} \overbrace{f N}^{\text{gradient}} - \overbrace{\left(V \cdot \frac{\partial C}{\partial s} \right) \frac{\partial V}{\partial s}}^{\text{advection}} - \overbrace{\frac{k+1}{t} V}^{\text{friction}} + \overbrace{\tau \frac{\partial^2 V}{\partial s^2}}^{\text{diffusion}},$$

where $\tau > 0$ represents a tunable diffusion coefficient. Large values of τ would give preferential treatment to coarse scale deformations of the evolving contour during the early stages of evolution, with finer scale deformations gradually folding in more and more as the contour converges toward a steady state configuration.

Such a coarse-to-fine behavior would be consistent with that of a Sobolev active contour. In fact, diffusion over a finite amount of time is similar to convolution with a smoothing kernel, which is indeed one way to relate the velocity field of a Sobolev active contour with the simple gradient field fN . As such, the incorporation of a diffusion term into the acceleration PDE is the closest and most direct way to endow the accelerated active contour with additional coarse-to-fine Sobolev active contour behaviors without directly employing Sobolev norms in the definition of the kinetic energy (which would require full linear operator inversion at every time step during the accelerated flow, just as in actual Sobolev gradient flows).

A key difference of such an added diffusion term, compared to Sobolev active contours, is that this smoothing process of the gradient field along the contour is carried out concurrently with the accelerated contour evolution itself, rather than statically at each separate time step. As such, if the diffusion coefficient τ is small enough to allow stable discretization of the PDE with the same time step dictated by the other first-order terms, then no additional computational cost is incurred. As the diffusion coefficient is increased, however, the discrete CFL conditions arising from the added second-order diffusion term will begin to dominate in the numerical implementation of the PDE and require smaller and smaller time steps. This could significantly increase the computational cost as more and more numerical iterations will be needed to simulate the same amount of accelerated flow time.

Given that a sufficiently small amount of diffusion costs essentially nothing in the PDE discretization, however, it doesn't make sense to ignore this benefit from an optimization standpoint. Methodical schemes guided purely by numerical considerations can be devised to add velocity diffusion coefficients that will maximally boost the regularity of the accelerated evolution with minimal or no added computational cost. Such *free gains* from small amounts of diffusion may be stretched the farthest by allowing variable diffusion coefficients which can be chosen based on evolving CFL conditions relevant to the PDE discretizations prior to considering the added diffusion terms.

3.5.2. Incorporating mass potential energy. An independent approach that would add spatial regularization to the acceleration process, again without imposing any added regularity to the converged result, would be to attach not only a kinetic energy term to the artificially attributed mass, but also an extra potential energy term \mathbf{U}_{mass} which favors a smoother evolution of the mass itself (and therefore of the object to which the auxiliary density function is attributed). This opens up a whole new design feature for accelerated PDEs which would allow us to incorporate coarse-to-fine evolution properties which are qualitatively similar to those of Sobolev gradient flows, but without the heavy computational cost. We foresee at least two criteria that should be satisfied when designing the mass potential energy term \mathbf{U}_{mass} .

1. The minimum achievable mass potential energy should be independent of the configuration of the original variable being optimized (for example, in the active contour case, it should be achievable for any possible contour shape) so that the final converged result, which will correspond to a locally minimal total potential energy, will not be influenced by the added mass potential energy term but only by the original potential energy term to be minimized. As such, the incorporation of \mathbf{U}_{mass} will affect only the accelerated evolution dynamics, without changing the original energy landscape.
2. The first variation $\delta\mathbf{U}_{mass}$ should not contain second- or higher-order derivatives of the density function ρ (nor of its flow velocity V) which would, like the diffusion strategy

described earlier, impose stronger discrete time step restrictions on the numerical discretization of the accelerated PDE system.

In order to work out a concrete example, we revisit the accelerated active contour model using the conserved flowable mass strategy outlined in section 3.2.3, in which we suggested that the evolving contour may be thought of as a moving container of fluid (the attributed mass variable), and that the fluid is pushed around by the moving container while also flowing within the container. If the fluid is compressible, then its density can vary during this evolution, otherwise, it must remain constant, which undermines the flexibility of this scheme compared to the simpler constant density scheme already developed beforehand in section 3.2.2. Yet we can still give physical intuition to the more flexible flowable mass model, even if we consider the mass as an incompressible fluid. We simply imagine that the fluid has a variable height at each point within its container (in this case, along the contour). This allows us to naturally define a potential energy for the mass configuration, by relating the density function ρ to the fluid height.

Using this fluid height model, we may construct the mass potential energy connected with an arc length increment ds along the curve by first noting that the associated mass differential is given by $dm = \rho ds$ and then equating the mass density ρ along the contour to a constant fluid density σ scaled by the local fluid height h . Given that the average height of the fluid column over ds would be $h/2$, we may write its potential energy as $\frac{h}{2} g dm$, where g represents a gravitational constant. Combining these relationships yields

$$d\mathbf{U} = \frac{g}{2\sigma} \rho^2 ds$$

which, if we choose $\sigma = 1$ (without any loss of generality since g can be chosen arbitrarily), gives the following expression for the mass potential energy:

$$g \int_C \frac{1}{2} \rho^2 ds.$$

However, while this satisfies our second criterion (its first variation will not involve second-order derivatives of ρ or of the flow velocity V), it fails our first criterion. To see this, note that the lowest potential energy mass distribution for a given curve (subject to the conservation constraint) is achieved by the constant height distribution $\rho = \frac{M}{L}$, where M denotes the constant total conserved mass and where $L = \int_C ds$ denotes the total arc length of the contour:

$$\min_{\rho} \left(g \int_C \frac{1}{2} \rho^2 ds \right) = \frac{g}{2} \frac{M^2}{L}.$$

From this expression, we can see that scaling this potential energy by the length of the curve will make the minimum achievable potential energy $\frac{g}{2} M^2$ become independent of the curve C . This leads to the following candidate for a mass potential energy which also satisfies our first criterion:

$$(3.15) \quad \mathbf{U}_{mass} = g L \int_C \frac{1}{2} \rho^2 ds.$$

Adding this to the purely contour based potential energy (section 3.2.1), which does not depend on the artificially added mass, and recomputing the Euler–Lagrange equations for

the generalized action integral (3.1) will yield a new system of accelerated PDEs in which the gradient forces influencing the acceleration will depend both on the mass distribution as well as the functional to be minimized. Since the minimum constant density mass potential can be achieved for any contour configuration, we know that at steady state, we will have a constant mass density. If we initialize with a constant mass density as well, then the acceleration dynamics will favor (but not constrain) moving the mass along evolution paths that keep the density spatially constant. Translations or uniform rescaling of the curve would therefore become preferential evolutions, just as for Sobolev active contours, especially with larger choices of the tunable gravitational constant g .

3.6. Incorporating stochastic acceleration terms. Finally, the accelerated PDE framework, unlike the gradient descent PDE framework, offers a numerical opportunity to introduce random noise into the evolution process without destroying the continuity of the evolution process nor of the evolving object. For example, in the active contour acceleration scheme (3.13), we could replace the added diffusion term suggested in section 3.5, with a stochastic term as follows:

$$\overbrace{\frac{\partial V}{\partial t}}^{\text{acceleration}} = \frac{\lambda k^2 t^{k-2}}{\rho} \overbrace{f N}^{\text{gradient}} - \overbrace{\left(V \cdot \frac{\partial C}{\partial s} \right) \frac{\partial V}{\partial s}}^{\text{advection}} - \overbrace{\frac{k+1}{t} V}^{\text{friction}} + \overbrace{\tau \mathcal{W}}^{\text{noise}}, \quad \frac{\partial C}{\partial t} = (V \cdot N)N,$$

where \mathcal{W} represents samples drawn from a random noise process and τ is a positive tunable coefficient (similar to the diffusion coefficient in section 3.5). Since the noise is added to the acceleration, it gets integrated twice in the construction of the updated contour (or surface) and therefore does not immediately interfere with the continuity nor the first-order differentiability of the evolving variable. As such, both the velocity V itself, as well as the unit normal N of the contour, remain continuous for the the coupled contour evolution equation. The contour therefore maintains regularity (at least short term). Furthermore, since upwind differencing methods are utilized in the numerical calculation of $\frac{\partial V}{\partial s}$ in the acceleration advection term, discontinuities in the first derivative of V do not pose a problem as only one-sided derivatives are required. In the case of shocks, a viscosity solution will be approximated by a proper discretization.

Adding random noise to a standard (nonaccelerated) gradient descent contour PDE,

$$\overbrace{\frac{\partial C}{\partial t}}^{\text{velocity}} = \overbrace{f N}^{\text{gradient}} + \overbrace{\tau \mathcal{W}}^{\text{noise}},$$

on the other hand, has never been a viable option since noise added directly to the velocity is integrated only once, which does not maintain continuity in the unit normal N of the evolving contour. As such, the contour would immediately become irregular. As such, accelerated PDEs open up a whole new avenue for the inclusion of stochastic terms (as often exploited in finite dimensional optimization problems) which offer an additional strategy for increased resistance to spurious or shallow local minimizers. The potential benefit of such a random noise term would be to provide a second and independent mechanism (beyond the acceleration itself) to perturb the optimization flow away from saddle points or shallow minimizers. Once

kinetic energy has been accumulated, the added benefit of such a term is likely to be negligible. However, unlucky initializations (assuming zero initial velocity) near local minimizers or saddle points, could benefit from a noise driven term in the early stages while momentum is just beginning to accumulate. Note that such a strategy is not the same as SGD, and should not be confused with the recently developed PDE methods in [13, 14] which specifically improve upon stochastic gradient descent techniques used in training deep neural networks.

4. Illustrative results. In this section we illustrate the performance improvements of reformulating an existing active contour model into the accelerated framework and demonstrate favorable performance improvements even when comparing it against recent alternative global strategies such as Chambolle–Pock. As the scope of this paper is not to propose or invent a particular active contour model, but rather an accelerated framework that can apply to variational models, we keep the 2D test images simple, such that popular binary region based active contour models (such as Chan–Vese) are well suited to the segmentation tasks. We demonstrate that these models, however, when lacking sufficient regularity (in this case the arc length penalty), become prone to getting trapped within unwanted local minimizers when implemented as standard gradient descent active contours. We show that even strategies such as Chambolle–Pock, which seek to minimize the global energy, still become numerically trapped within local minimizers when used with matching regularity. And while these alternative global minimizers can perform admirably on special classes of binary region based active contours they are not extendable with the same generality as the PDE acceleration framework presented here. We will see in these illustrative examples that simply applying the contour acceleration is sufficient to fix the sensitivity to local minimizers, drastically speed up the convergence of the region based active contours, and without the need to abandon the active contour framework in favor of less generalizable global convex optimization methods. We also provide example extensions of the acceleration framework for nonconvex problems, particularly the case of variational 3D reconstruction, where we show not only a dramatic speedup in terms of runtime but also a better converged result for both toy and real data.

4.1. Acceleration versus gradient descent. In Figure 3 we see three different initial contour placements (top, middle, bottom) evolving from left-to-right via the gradient flow PDE (2.2). Each gets trapped within a different local minimizer due to noise, all of which lie very far away from the desired much deeper minimizer along the rectangular boundary. Of course, stronger regularizing terms could be added to the active contour energy functional to impose smoothness on the contour, thereby making it resistant to noise. However, the point of this experiment was to create an energy landscape littered with literally tens of thousands (perhaps even hundreds of thousands) of local minimizers in order to demonstrate the effects of acceleration. Furthermore, stronger regularization would sacrifice the ability to capture the sharp corners of the rectangle and increase the computational cost due to smaller resulting step size constraints in the PDE discretization.

We avoid both of these sacrifices by instead using the exact same active contour force f within the accelerated PDE system (3.8). In Figure 4, we see the effect of applying an accelerated contour evolution scheme with the same initial contour placements and same energy functional (no additional regularizing terms). In all three cases, the accelerated PDE system pushes the contour past the noise, driving it toward a more robust minimum along the rectangle edge.

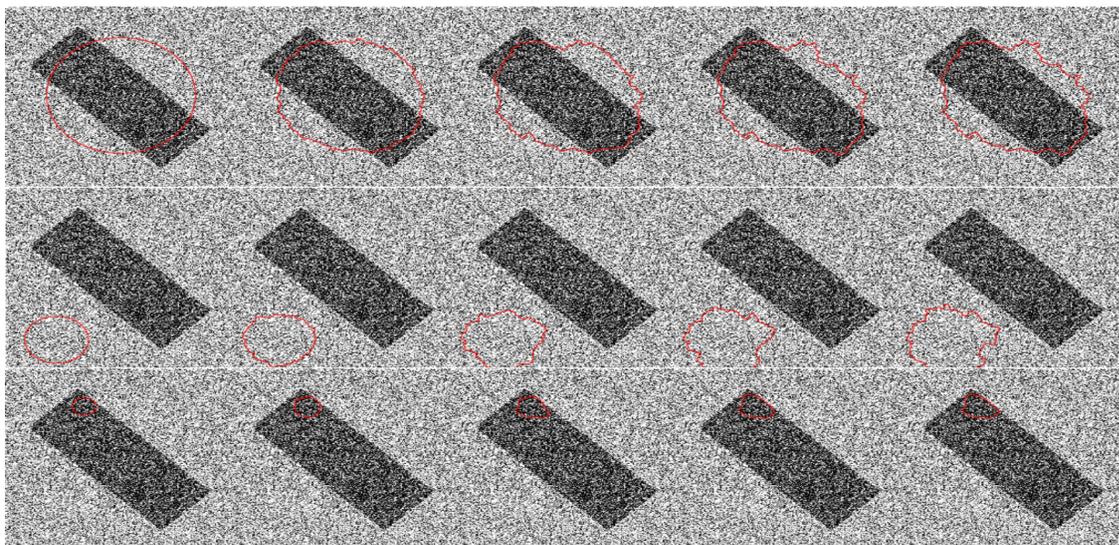


Figure 3. *Different initial contours flowing into local energy minimizers.*

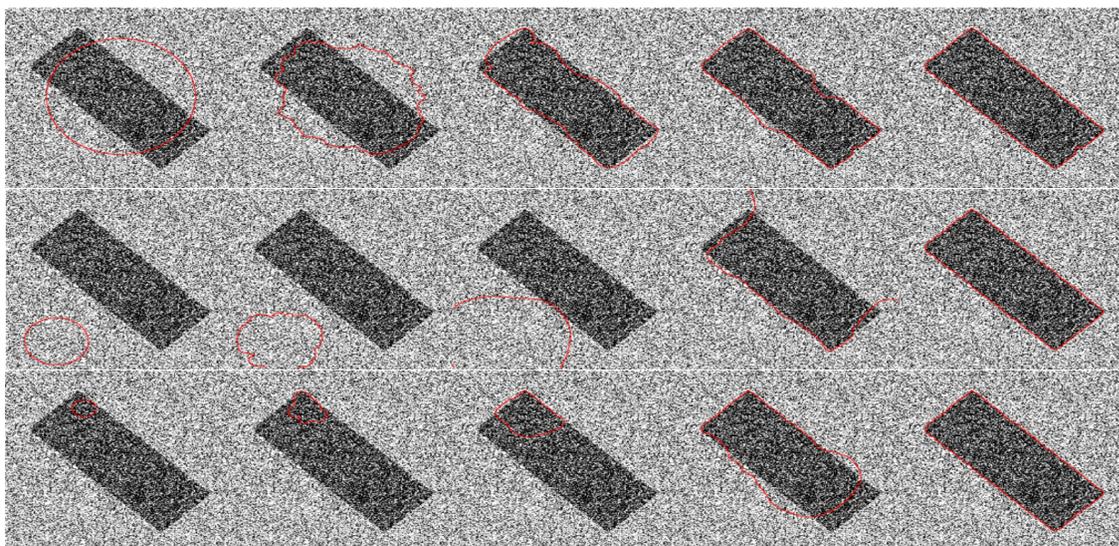


Figure 4. *Accelerated active contours flowing past local minima.*

In Figure 5 we see this same dramatic difference on a real seismographic image where we attempt to use an active contour to pull out the rather noisy “core” of the recorded seismograph line. Along the left column we see four different initial contour placements, where the first three elliptical initializations, which are far from the desired segmented result, pose a considerable challenge to a classical gradient descent active contour. Minimal regularization is allowed here given the spikey nature of the signal, at least in cases where we wish to capture this fine scale level of detail.

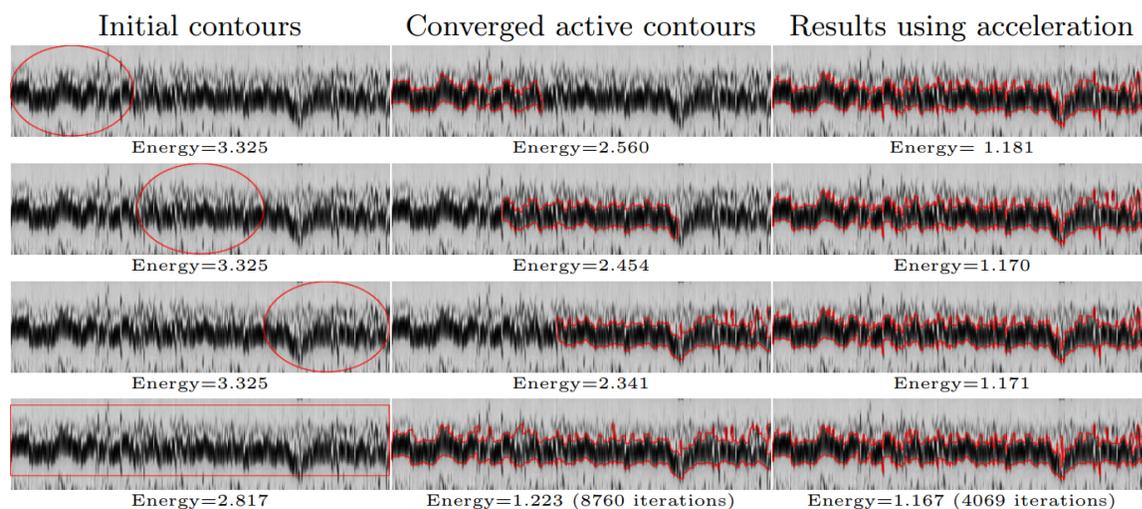


Figure 5. Nonaccelerated (middle) versus accelerated (right) active contour results for the same four initializations (left) on a seismograph image. Cost functional values underneath.

In the middle column, we see the converged active contour results based on the standard gradient flow version of the evolution given by (2.2). Only in the last (bottom) case, is the segmented result reasonable.

In the last column, we see the converged result of the same active contour energy E and force f evolved using the accelerated PDE system (3.8). While there are very subtle differences in the final results (as can be see by the slight differences in the converged energy value), all four are nonetheless reasonable now even from the first three challenging initial contour placements.

4.2. Comparison with convex relaxation. In Table 1, we compare our method, an accelerated version of active contours (ACs) to a global convex energy minimization method Chambolle/Pock (CP) [8], and find comparable robustness to global methods but with a significant computational savings. We choose the regularity such that standard ACs converge to a local minima (not the global) over multiple different initializations, so that a better method is required to optimize the energy. The regularity is also chosen with the performance of CP in mind for the comparison, as CP also requires a sufficiently high regularity, although lower than standard ACs, to segment the region.

We run all AC and CP experiments to convergence and measure the computational time and final energy for 3 different initial segments at 4 different resolutions. The experiments were a square close to the desired segmentation—“near square”; a square far from the desired segmentation—“far square”; and a binary threshold of the image—“threshold mask.” In Figure 1 we present a scaled down noisy binary image of resolution 1120×1120 with the final segmentation s for both methods. Performance results are provided in Table 1. This comparison shows that our method consistently obtains comparable local optima over different initializations, similar to CP, but with less computational time. Furthermore, our method applies more generally to nonconvex problems, where we would expect similar robustness in

Table 1

[Left]: PDE acceleration (AC) offers a comparable level of robustness to initialization as global convex CP in lower computational time. (Right): Visual comparison for the results with greatest energy difference in CP and AC shows that the energy differences are nearly in-perceptible.

	Time CP (sec)	Time AC (sec)	Energy AC	Energy CP
Near Square				
Res: 280 x 280	0.176	0.042	5.1960E+08	5.1864E+08
Res: 560 x 560	2.34	0.11	5.2120E+08	5.0995E+08
Res: 1120 x 1220	20.009	1.167	5.2120E+08	5.0726E+08
Res: 2240 x 2240	159.76	14.43	5.2016E+08	5.0563E+08
Threshold Mask				
Res: 280 x 280	0.174	0.064	5.1680E+08	5.2250E+08
Res: 560 x 560	2.24	0.204	5.1680E+08	5.0918E+08
Res: 1120 x 1120	20.438	1.055	5.1569E+08	5.0714E+08
Res: 2240 x 2240	159.178	14.606	5.2016E+08	5.0516E+08
Far Square				
Res: 280 x 280	0.771	0.278	5.1980E+08	5.1860E+08
Res: 560 x 560	8.726	1.43	5.1980E+08	5.0900E+08
Res: 1120 x 1120	91.197	14.65	5.2067E+08	5.0691E+08
Res: 2240 x 2240	772.342	69.48	5.2016E+08	5.0609E+08

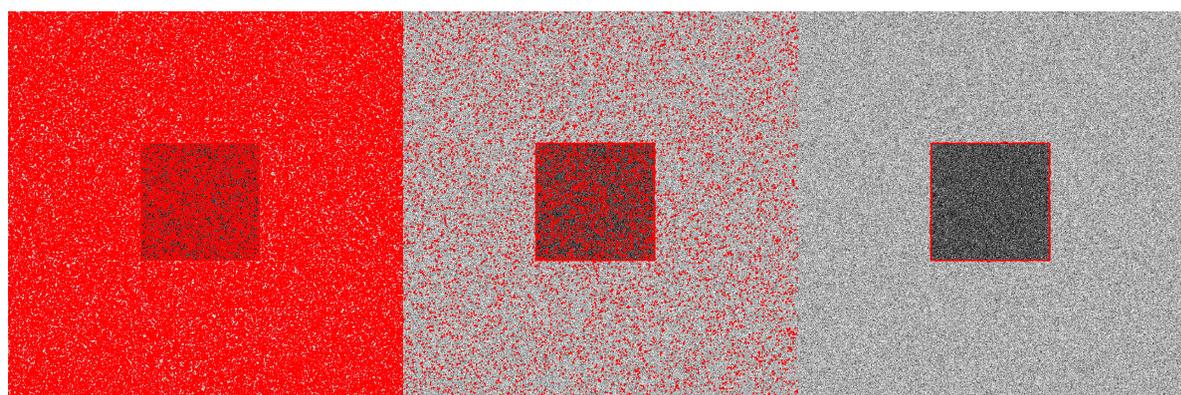
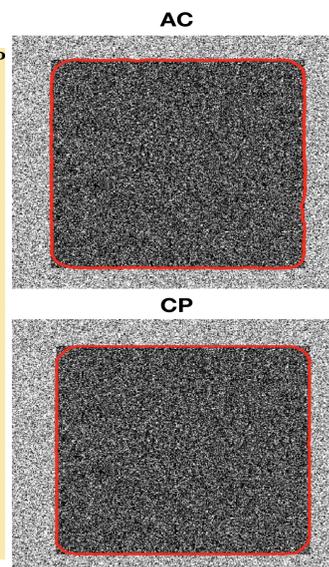


Figure 6. (Left): Initial threshold initialization of noisy square; (middle): Converged segmentation for CP; (right): Converged segmentation for accelerated AC.

our method, and where CP is not as applicable; see 3D stereoscopic segmentation experiments in section 4.3.

In Figure 6 we present the case of a segmentation based on an initial threshold of the image; this is a similar experiment to the threshold mask discussed earlier, but with a significantly lower arc length penalty than those used in Table 1. Here we show another advantage of the accelerated ACs method in needing less regularization for segmentation. In Figure 6 and Table 2 the numerical implementation of CP becomes trapped whereas the accelerated version of ACs is able to push past the local minimizers in the image.

Table 2

Performance numbers for threshold segmentation CP versus accelerated ACs for Figure 6.

Image resolution	Arc penalty	Iterations	CP energy	AC energy
1120 x 1120	1.00E+4	600	8.14E+9	8.07E+9

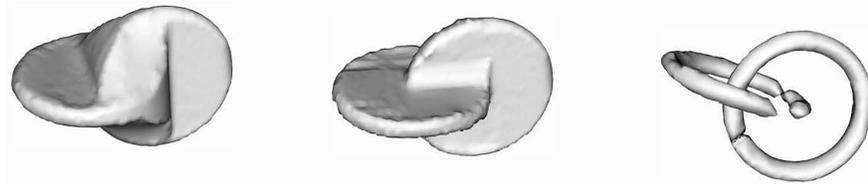


Figure 7. Three double tori reconstructions using gradient descent stereoscopic segmentation. (Left) torus used 16 images and 0 area penalty. (Middle) torus used 18 images and a moderate area penalty of 150. (Right) torus used 18 images and a high area penalty of 500. Notice the modest feature loss in the final image.

4.3. Application to nonconvex problems. For the case of nonconvex problems, we extended the gradient descent based stereoscopic segmentation from [61] to an accelerated version using the framework presented in this manuscript. We chose this model because of its extreme nonconvexity. The primary reason for this is that the projections in stereoscopic segmentation must be applied to both the occlusion boundaries and the evolving structure. The gradient descent optimization used in this stereoscopic segmentation model is also prone to getting trapped in local minima and demonstrates the potential benefits of PDE acceleration.

For the first experiment, we consider the case of a double torus with an initial bounding region that completely covers both torus' holes. In this synthetic experiment (see Figure 7), the gradient descent stereoscopic segmentation gets trapped in a local minimum and is unable to push through the holes. Adding additional camera views and some regularity does improve the results but with large local minimizers still remaining. While adding a much harsher area penalty can remove the local minimizers it does so at the expense of image features.

A similar experiment by Kolev, Brox, and Cremers [26] compares a double torus reconstruction using stereoscopic segmentation against a convex probabilistic fusion scheme that combines probable 3D shapes and observed color information. In [26] the authors astutely note that while the approximation in [63, 61] is more faithful it is not globally optimizable. However, by extending the nonconvex approach in [61] using PDE acceleration we are able to push through the local minimizers without the use of additional constraints on the geometry or even additional regularization.

Comparing the results in Figure 8, the accelerated PDE method is able to push through the local minimizers with ease and not only achieves a far better result but also converges far faster than the gradient method. A numeric performance comparison between the two methods is provided in Table 3.

The second reconstruction we present is of a wooden horse using 32 different 640×480 resolution images; see Figure 9. Comparing the gradient descent (bottom) and accelerated PDE (top) schemes, we demonstrate faster convergence for the accelerated PDE scheme as well as better robustness to local minimizers. Unlike in the gradient PDE method we are able

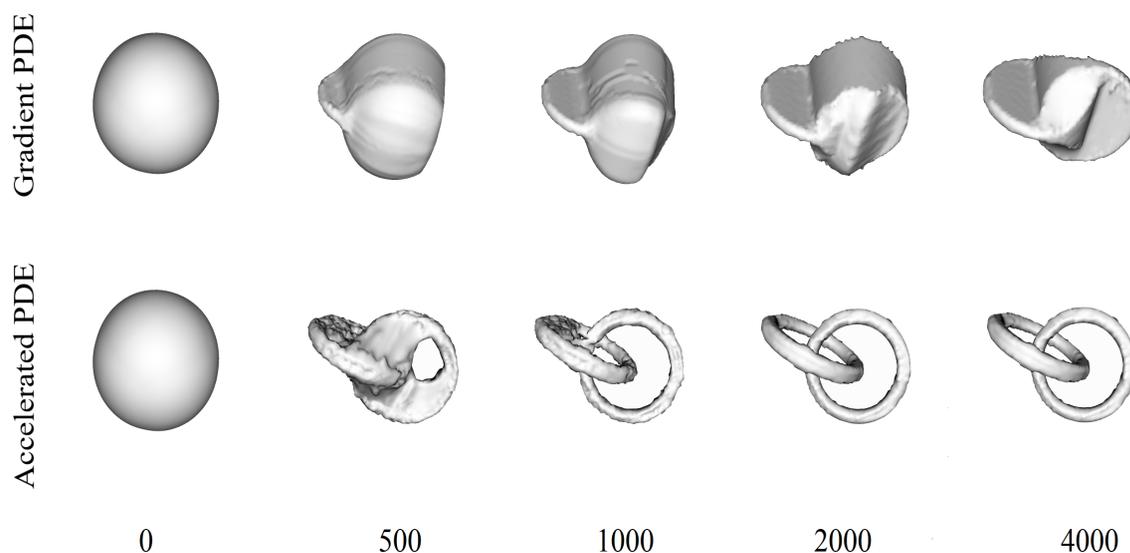


Figure 8. Comparison of stereoscopic segmentation for gradient PDE method versus accelerated PDE method at different iteration steps 0, 500, 1000, 2000, and 4000 iterations, respectively. The gradient method gets trapped by local minimizers, in this case the bounds of the torus, and is unable to form the holes without heavy regularization. Note that the initial start for both methods is the ellipsoid. The area penalty was fixed at 0 and this experiment used 16 image views of the tori for the reconstructions.

Table 3

Performance benchmark of gradient versus accelerated method; computations were done on an Intel 6 Core i7-5930K.

	Iterations to converge:	Runtime (sec):	Final energy
Gradient descent	3653	196.44	2.09E+10
PDE acceleration	1630	66.16	2.90E+09

Table 4

Performance numbers for gradient descent versus PDE acceleration of variational 3D reconstruction. Note that increase in energy is due to the methods not capturing the entire horse volume.

Area penalty	Gradient descent			Acceleration		
	Iterations	Runtime (sec)	Final energy	Iterations	Runtime (sec)	Final energy
300	751	27.00	1.099E+10	205	8.307	9.381E+09

to capture the entire horse volume and do not lose smaller features such as the horse's thin legs.

Imposing a high area penalty on the horse reconstruction creates local minimizers within the horse's narrow legs. Gradient descent stereoscopic segmentation then struggles to push down the legs and reconstruct the entire horse volume. The accelerated version progresses far faster and does not suffer a penalty even under the same heavy regularization.

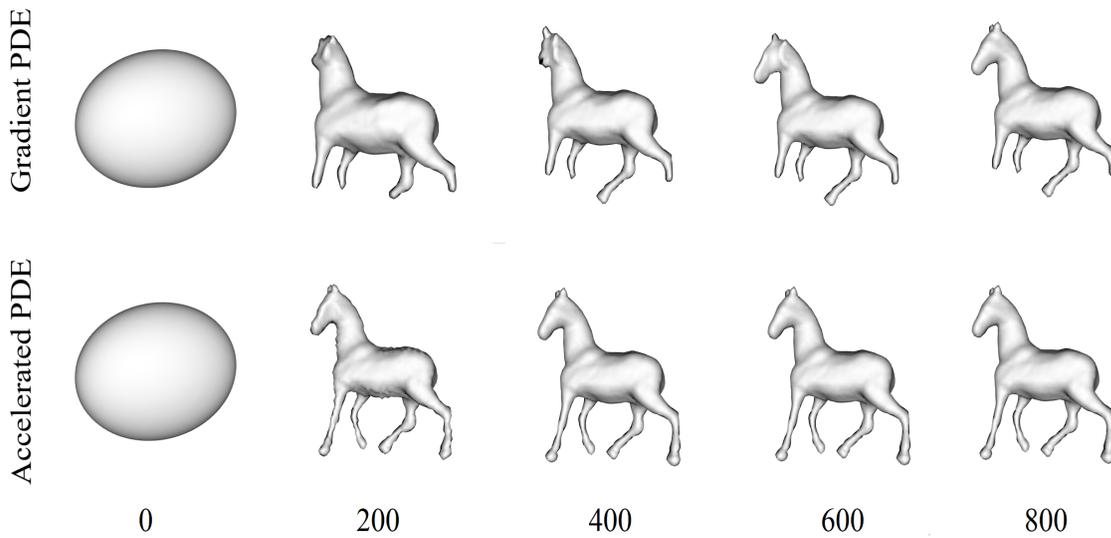


Figure 9. (Top): Gradient descent driven variational 3D reconstruction gets trapped in local minima when strongly regularized. (Bottom): PDE acceleration is able to successfully shoot past local minimizers under a higher area penalty. Note average convergence time for PDE acceleration is almost three times faster than gradient descent. Iterations are given at the bottom with performance results found in Table 4 on the previous page.

5. Appendix: Derivations of various numbered equations.

5.1. Calculation of (3.3). Differentiating (3.2) with respect to the arc length parameter s yields

$$\frac{\partial^2 C}{\partial s \partial t} = \frac{\partial \alpha}{\partial s} T + \underbrace{\alpha \frac{\partial T}{\partial s}}_{\kappa N} + \frac{\partial \beta}{\partial s} N + \beta \underbrace{\frac{\partial N}{\partial s}}_{-\kappa T} = \left(\frac{\partial \alpha}{\partial s} - \beta \kappa \right) T + \left(\frac{\partial \beta}{\partial s} + \alpha \kappa \right) N$$

and differentiating $T = \frac{\partial C}{\partial s}$ yields

$$\begin{aligned} \frac{\partial T}{\partial t} &= \frac{\partial}{\partial t} \frac{\partial C}{\partial s} = \frac{\partial}{\partial t} \left(\frac{\frac{\partial C}{\partial p}}{\left\| \frac{\partial C}{\partial p} \right\|} \right) = \frac{\frac{\partial^2 C}{\partial t \partial p}}{\left\| \frac{\partial C}{\partial p} \right\|} - \frac{\frac{\partial C}{\partial p}}{\left\| \frac{\partial C}{\partial p} \right\|^2} \frac{\partial}{\partial t} \left\| \frac{\partial C}{\partial p} \right\| = \frac{\frac{\partial^2 C}{\partial p \partial t}}{\left\| \frac{\partial C}{\partial p} \right\|} - \frac{\frac{\partial C}{\partial p}}{\left\| \frac{\partial C}{\partial p} \right\|^2} \frac{\partial^2 C}{\partial p \partial t} \cdot \frac{\partial C}{\partial p} \\ &= \frac{\partial^2 C}{\partial s \partial t} - T \left(\frac{\partial^2 C}{\partial s \partial t} \cdot T \right) = \left(\frac{\partial^2 C}{\partial s \partial t} \cdot N \right) N = \left(\frac{\partial \beta}{\partial s} + \alpha \kappa \right) N \end{aligned}$$

which gives the first part of (3.3) with the second part due to the rotation relationship between T and N .

5.2. Calculation of (3.9). Letting $C(p, t)$ denote a parameterization of the evolving curve C with a time-independent spatial parameter p and with s denoting the time-dependent arc length parameter we compute (ignoring temporary boundary terms when applying integration

by parts and assuming a closed curve so that spatial boundary terms cancel)

$$\begin{aligned}
& \delta \int_0^1 \frac{t^{k+1}}{k} \left(\mathbf{T} - \lambda k^2 t^{k-2} \mathbf{U} \right) dt = \delta \int_0^1 \left(\int_0^1 \frac{1}{2} \frac{t^{k+1}}{k} \rho \frac{\partial C}{\partial t} \cdot \frac{\partial C}{\partial t} ds - \lambda k t^{2k-1} \mathbf{U} \right) dt \\
&= \int_0^1 \delta \left(\int_0^1 \frac{1}{2} \frac{t^{k+1}}{k} \rho \frac{\partial C}{\partial t} \cdot \frac{\partial C}{\partial t} \left\| \frac{\partial C}{\partial p} \right\| dp - \lambda k t^{2k-1} \mathbf{U} \right) dt \\
&= \int_0^1 \frac{\rho}{k} \left(\int_0^1 \frac{1}{2} t^{k+1} \left(\frac{\partial C}{\partial t} \cdot \delta \frac{\partial C}{\partial t} \left\| \frac{\partial C}{\partial p} \right\| + \frac{1}{2} \left\| \frac{\partial C}{\partial t} \right\|^2 \delta \left\| \frac{\partial C}{\partial p} \right\| \right) dp - \frac{\lambda k^2 t^{2k-1}}{\rho} \delta \mathbf{U} \right) dt \\
&= \int_0^1 \frac{\rho}{k} \left(\int_0^1 -\frac{\partial}{\partial t} \left(t^{k+1} \frac{\partial C}{\partial t} \left\| \frac{\partial C}{\partial p} \right\| \right) \right. \\
&\quad \cdot \delta C + \frac{1}{2} t^{k+1} \left\| \frac{\partial C}{\partial t} \right\|^2 \delta \frac{\partial C}{\partial p} \cdot \frac{\partial C}{\partial s} dp + \frac{\lambda k^2 t^{2k-1}}{\rho} \int_C f (\delta C \cdot N) ds \left. \right) dt \\
&= \int_0^1 \frac{\rho}{k} \left(\int_0^1 -t^{k+1} \frac{\partial^2 C}{\partial t^2} \left\| \frac{\partial C}{\partial p} \right\| - (k+1) t^k \frac{\partial C}{\partial t} \left\| \frac{\partial C}{\partial p} \right\| \right. \\
&\quad \left. - t^{k+1} \frac{\partial C}{\partial t} \frac{\partial}{\partial t} \left\| \frac{\partial C}{\partial p} \right\| - \frac{1}{2} t^{k+1} \frac{\partial}{\partial p} \left(\left\| \frac{\partial C}{\partial t} \right\|^2 \frac{\partial C}{\partial s} \right) \right) \cdot \delta C dp + \frac{\lambda k^2 t^{2k-1}}{\rho} \int_C f N \cdot \delta C ds \left. \right) dt \\
&= \int_0^1 \frac{t^{k+1}}{k} \rho \left(\int_0^1 \left(-\frac{\partial^2 C}{\partial t^2} \left\| \frac{\partial C}{\partial p} \right\| - \frac{k+1}{t} \frac{\partial C}{\partial t} \left\| \frac{\partial C}{\partial p} \right\| - \frac{\partial C}{\partial t} \left(\frac{\partial^2 C}{\partial p \partial t} \cdot \frac{\partial C}{\partial s} \right) \right. \right. \\
&\quad \left. \left. - \frac{1}{2} \frac{\partial}{\partial p} \left(\left\| \frac{\partial C}{\partial t} \right\|^2 \frac{\partial C}{\partial s} \right) \right) \cdot \delta C dp + \frac{\lambda k^2 t^{k-2}}{\rho} \int_C f N \cdot \delta C ds \right) dt \\
&= \int_0^1 \frac{t^{k+1}}{k} \rho \int_C \underbrace{\left(-\frac{\partial^2 C}{\partial t^2} - \frac{k+1}{t} \frac{\partial C}{\partial t} - \left(\frac{\partial^2 C}{\partial s \partial t} \cdot \frac{\partial C}{\partial s} \right) \frac{\partial C}{\partial t} - \frac{\partial}{\partial s} \left(\frac{1}{2} \left\| \frac{\partial C}{\partial t} \right\|^2 \frac{\partial C}{\partial s} \right) + \frac{\lambda k^2 t^{k-2}}{\rho} f N \right)}_{\text{Set to zero for Euler-Lagrange equation}} \\
&\quad \cdot \delta C ds dt.
\end{aligned}$$

5.3. Calculation of (3.7). Decomposing the acceleration C_{tt} into tangential and normal components yields

$$\begin{aligned}
\frac{\partial^2 C}{\partial t^2} &= -\frac{k+1}{t} \frac{\partial C}{\partial t} - \left(\frac{\partial^2 C}{\partial s \partial t} \cdot \frac{\partial C}{\partial s} \right) \frac{\partial C}{\partial t} - \left(\frac{\partial^2 C}{\partial s \partial t} \cdot \frac{\partial C}{\partial t} \right) \frac{\partial C}{\partial s} - \frac{1}{2} \left\| \frac{\partial C}{\partial t} \right\|^2 \frac{\partial^2 C}{\partial s^2} + \frac{\lambda k^2 t^{k-2} f}{\rho} N \\
&= -\frac{k+1}{t} \frac{\partial C}{\partial t} - \left(\frac{\partial \alpha}{\partial s} - \beta \kappa \right) \frac{\partial C}{\partial t} - \left(\alpha \frac{\partial \alpha}{\partial s} + \beta \frac{\partial \beta}{\partial s} \right) T - \frac{\alpha^2 + \beta^2}{2} \kappa N + \frac{\lambda k^2 t^{k-2} f}{\rho} N, \\
\frac{\partial^2 C}{\partial t^2} \cdot T &= -\left(\frac{k+1}{t} + \frac{\partial \alpha}{\partial s} - \beta \kappa \right) \alpha - \left(\alpha \frac{\partial \alpha}{\partial s} + \beta \frac{\partial \beta}{\partial s} \right) = -\left(\frac{k+1}{t} + 2 \frac{\partial \alpha}{\partial s} - \beta \kappa \right) \alpha - \beta \frac{\partial \beta}{\partial s}, \\
\frac{\partial^2 C}{\partial t^2} \cdot N &= -\left(\frac{k+1}{t} + \frac{\partial \alpha}{\partial s} - \beta \kappa \right) \beta - \frac{\alpha^2 + \beta^2}{2} \kappa + \frac{\lambda k^2 t^{k-2} f}{\rho}.
\end{aligned}$$

Now inserting these acceleration components into (3.5) yields (3.8):

$$\begin{aligned} \frac{\partial \alpha}{\partial t} &= - \underbrace{\left(\frac{k+1}{t} + 2 \frac{\partial \alpha}{\partial s} - \beta \kappa \right)}_{\frac{\partial^2 C}{\partial t^2} \cdot T} \alpha - \beta \frac{\partial \beta}{\partial s} + \beta \left(\frac{\partial \beta}{\partial s} + \alpha \kappa \right) = \left(-\frac{k+1}{t} - 2 \frac{\partial \alpha}{\partial s} + 2\beta \kappa \right) \alpha, \\ \frac{\partial \beta}{\partial t} &= - \underbrace{\left(\frac{k+1}{t} + \frac{\partial \alpha}{\partial s} - \beta \kappa \right)}_{\frac{\partial^2 C}{\partial t^2} \cdot N} \beta - \frac{\alpha^2 + \beta^2}{2} \kappa + \frac{\lambda k^2 t^{k-2} f}{\rho} - \alpha \left(\frac{\partial \beta}{\partial s} + \alpha \kappa \right), \\ &= -\frac{k+1}{t} \beta - \frac{\partial}{\partial s} (\alpha \beta) + \left(\frac{1}{2} \beta^2 - \frac{3}{2} \alpha^2 \right) \kappa + \frac{\lambda k^2 t^{k-2} f}{\rho}. \end{aligned}$$

5.4. Calculation of (3.9). Assuming we represent the evolving curve $C(p, t)$ as the zero level set of an evolving function $\psi(x, t)$ and letting $\hat{\beta}(x, t)$ denote an evolving spatial extension of the evolving normal speed function $\beta(p, t)$ along curve, then we have

$$\psi(C(p, t), t) = 0 \quad \text{and} \quad \hat{\beta}(C(p, t), t) = \beta(p, t).$$

Differentiating with respect to t yields

$$\frac{\partial \psi}{\partial t} + \nabla \psi \cdot \frac{\partial C}{\partial t} = 0 \quad \text{and} \quad \frac{\partial \hat{\beta}}{\partial t} + \nabla \hat{\beta} \cdot \frac{\partial C}{\partial t} = \frac{\partial \beta}{\partial t}.$$

Extending the contour evolution $\frac{\partial C}{\partial t} = \beta N$ to other level sets as $\hat{\beta} \hat{N}$, where $\hat{N} = -\frac{\nabla \psi}{\|\nabla \psi\|}$ (noting that this convention for the extension of the inward unit normal requires that the level set function be negative inside the contour and positive outside), yields

$$\frac{\partial \psi}{\partial t} = \hat{\beta} \|\nabla \psi\| \quad \text{and} \quad \frac{\partial \hat{\beta}}{\partial t} = \frac{\partial \beta}{\partial t} + \nabla \hat{\beta} \cdot \frac{\hat{\beta} \nabla \psi}{\|\nabla \psi\|}$$

which, after substitution of $\frac{\partial \beta}{\partial t}$ using (3.8) results in the level set version of the system in (3.9).

5.5. Calculation of (3.12). Let us introduce, along with the mass density ρ and its internal flow speed v with respect to the arc length parameter s , corresponding variables for the mass density $\mu(p, t)$ and internal flow speed $\xi(p, t)$ with respect to a time-independent contour parameter p . These pairs of densities and internal flow speeds are related to each other through the parameterization speed $\|C_p\|$ of the contour as follows:

$$(5.1) \quad \mu = \rho \|C_p\| \quad \text{and} \quad v = \xi \|C_p\| \quad (\text{with matching flux expressions } \mu \xi = \rho v).$$

Differentiating with respect to t , yields the following relationships between the density and flow speed evolution as well:

$$(5.2) \quad \mu_t - \mu C_{ts} \cdot C_s = \rho_t \|C_p\| \quad \text{and} \quad v_t - v C_{ts} \cdot C_s = \xi_t \|C_p\|'.$$

Applying these substitutions to the kinetic energy (3.10) and continuity constraint (3.11) yields

$$\mathbf{T} = \int_0^1 \frac{1}{2} \mu \|C_t + \xi C_p\|^2 dp \quad \text{with mass continuity constraint} \quad \mu_t + (\mu \xi)_p = 0.$$

We plug this into the generalized action integral (3.1) with a Lagrange multiplier function $\lambda(p, t)$ and compute the first variation:

$$\begin{aligned} & \delta \int_0^1 \frac{t^{k+1}}{k} \left(\mathbf{T} - \lambda k^2 t^{k-2} \mathbf{U} \right) + \int_0^1 \lambda (\mu_t + (\mu \xi)_p) dp dt \\ &= \int_0^1 \delta \int_0^1 \frac{1}{2} \frac{t^{k+1}}{k} \mu \|C_t + \xi C_p\|^2 + \lambda (\mu_t + (\mu \xi)_p) dp - \lambda k t^{2k-1} \delta \mathbf{U} dt \\ &= \int_0^1 \int_0^1 \frac{1}{2} \frac{t^{k+1}}{k} \|C_t + \xi C_p\|^2 \delta \mu + \frac{t^{k+1}}{k} \mu (C_t + \xi C_p) \cdot \delta (C_t + \xi C_p) \\ & \quad + \underbrace{(\mu_t + (\mu \xi)_p)}_{=0} \delta \lambda + \lambda \delta (\mu_t + (\mu \xi)_p) + \lambda k t^{2k-1} f (\delta C \cdot N) \|C_p\| dp dt \\ &= \int_0^1 \int_0^1 \frac{1}{2} \frac{t^{k+1}}{k} \|C_t + \xi C_p\|^2 \delta \mu + \frac{t^{k+1}}{k} \mu (C_t + \xi C_p) \cdot C_p \delta \xi - \lambda_t \delta \mu - \lambda_p \delta (\mu \xi) \\ & \quad - \left(\frac{t^{k+1}}{k} \mu (C_t + \xi C_p) \right)_t \cdot \delta C - \left(\frac{t^{k+1}}{k} \mu \xi (C_t + \xi C_p) \right)_p \cdot \delta C + \lambda k t^{2k-1} f N \cdot \delta C \|C_p\| dp dt \\ &= \int_0^1 \int_0^1 \left(\frac{1}{2} \frac{t^{k+1}}{k} \|C_t + \xi C_p\|^2 - \lambda_t - \lambda_p \xi \right) \delta \mu + \mu \left(\frac{t^{k+1}}{k} (C_t + \xi C_p) \cdot C_p - \lambda_p \right) \delta \xi \\ & \quad - \frac{t^{k+1}}{k} \mu \left(\underbrace{(\mu_t + (\mu \xi)_p)}_{=0} \frac{C_t + \xi C_p}{\mu} + (C_t + \xi C_p)_t + \xi (C_t + \xi C_p)_p \right. \\ & \quad \left. + \frac{k+1}{t} (C_t + \xi C_p) - \frac{\lambda k^2 t^{k-2} f N}{\rho} \right) \cdot \delta C dp dt. \end{aligned}$$

The optimality conditions with respect to variations $\delta \xi$ and $\delta \mu$, respectively, yield

$$\lambda_p = \frac{t^{k+1}}{k} (\xi \|C_p\|^2 + C_t \cdot C_p) \quad \text{and} \quad \lambda_t = \frac{1}{2} \frac{t^{k+1}}{k} \|C_t + \xi C_p\|^2 - \xi \lambda_p$$

which, when combined, give the following evolution for the Lagrange multiplier:

$$\begin{aligned} \lambda_t &= \frac{t^{k+1}}{k} \left(\frac{1}{2} \|C_t + \xi C_p\|^2 - (\|\xi C_p\|^2 + C_t \cdot \xi C_p) \right) \\ &= \frac{t^{k+1}}{k} \left(\frac{1}{2} \|C_t + v C_s\|^2 - (\|v C_s\|^2 + C_t \cdot v C_s) \right) = \frac{1}{2} \frac{t^{k+1}}{k} (\|C_t\|^2 - v^2). \end{aligned}$$

We eliminate the Lagrange multiplier by equating λ_{tp} and λ_{pt} to obtain the following internal

flow speed evolution:

$$\begin{aligned}
0 &= \lambda_{tp} - \lambda_{pt} = \frac{1}{2} \frac{t^{k+1}}{k} (\|C_t\|^2 - v^2)_p - \left(\frac{t^{k+1}}{k} (C_t + \xi C_p) \cdot C_p \right)_t, \\
0 &= \frac{t^{k+1}}{k} \left(\underbrace{C_t \cdot C_{tp}}_{\text{cancel}} - vv_p - C_{tt} \cdot C_p - \xi_t C_p \cdot C_p - 2\xi C_{tp} \cdot C_p - \underbrace{C_t \cdot C_{tp}}_{\text{cancel}} - \frac{k+1}{t} (C_t + \xi C_p) \cdot C_p \right), \\
0 &= \underbrace{\|C_p\|}_{\text{drop}} \left(-vv_s - C_{tt} \cdot C_s - \underbrace{(\xi_t \|C_p\| + v C_{ts} \cdot C_s)}_{v_t \text{ by using (5.2)}} - v C_{ts} \cdot C_s - \frac{k+1}{t} (C_t + v C_s) \cdot C_s \right), \\
v_t &= - \left(C_{tt} + v (C_t + v C_s)_s + \frac{k+1}{t} (C_t + v C_s) \right) \cdot C_s \\
&= - \left(C_{tt} + v C_{ts} + \frac{k+1}{t} C_t \right) \cdot C_s - \left(v_s + \frac{k+1}{t} \right) v.
\end{aligned}$$

Finally, the optimality condition with respect to the curve perturbation δC yields the following acceleration equation for the contour

$$\begin{aligned}
0 &= C_{tt} + \underbrace{\xi_t \|C_p\|}_{v_t - v C_{ts} \cdot C_s} C_s + v C_{ts} + v (C_t + v C_s)_s + \frac{k+1}{t} (C_t + v C_s) - \frac{\lambda k^2 t^{k-2} f}{\rho} N, \\
0 &= C_{tt} + (v_t - v C_{ts} \cdot C_s) C_s + v C_{ts} + v (C_t + v C_s)_s + \frac{k+1}{t} (C_t + v C_s) - \frac{\lambda k^2 t^{k-2} f}{\rho} N, \\
0 &= \underbrace{\left(C_{tt} + v (C_t + v C_s)_s + \frac{k+1}{t} (C_t + v C_s) + v C_{ts} \right)}_{\text{some vector}} \\
&\quad - \underbrace{\left(\left(C_{tt} + v (C_t + v C_s)_s + \frac{k+1}{t} (C_t + v C_s) + v C_{ts} \right) \cdot C_s \right)}_{\text{its tangential component}} C_s - \frac{\lambda k^2 t^{k-2} f}{\rho} N, \\
0 &= \underbrace{\left(C_{tt} + v (C_t + v C_s)_s + \frac{k+1}{t} (C_t + v C_s) + v C_{ts} \right)}_{\text{its normal projection}} \cdot N - \frac{\lambda k^2 t^{k-2} f}{\rho} \\
&= \left(C_{tt} + 2v C_{ts} + v^2 C_{ss} + \frac{k+1}{t} C_t \right) \cdot N - \frac{\lambda k^2 t^{k-2} f}{\rho}.
\end{aligned}$$

5.6. Calculation of (3.13). Plugging the normal and tangential (unconstrained) acceleration components from (3.12) into (3.5) yields

$$\begin{aligned}\alpha_t &= \underbrace{C_{tt} \cdot T}_{\text{free}} + \beta(\beta_s + \alpha\kappa) \\ \beta_t &= \underbrace{\left(-2v \frac{C_{ts} \cdot N}{(\beta_s + \alpha\kappa)} - v^2 \frac{C_{ss} \cdot N}{\kappa} - \frac{k+1}{t} \frac{C_t \cdot N}{\beta} + \lambda k^2 t^{k-2} \frac{f}{\rho}\right)}_{C_{tt} \cdot N \text{ (from (3.12))}} - \alpha(\beta_s + \alpha\kappa), \\ &= -(2v + \alpha)\beta_s - (\alpha + v)^2 \kappa - \frac{k+1}{t} \beta + \lambda k^2 t^{k-2} \frac{f}{\rho}, \\ v_t &= -C_{tt} \cdot T - v \frac{C_{ts} \cdot T}{(\alpha_s - \beta\kappa)} - \frac{k+1}{t} \left(\frac{C_t \cdot T}{\alpha} + v\right) - vv_s \quad (\text{from second part of (3.12)}), \\ \alpha_t + v_t &= (C_{tt} \cdot T + \beta\beta_s + \alpha\beta\kappa) + \left(-C_{tt} \cdot T - v(\alpha_s - \beta\kappa) - vv_s - \frac{k+1}{t}(\alpha + v)\right), \\ (\alpha + v)_t &= \beta\beta_s - (\alpha_s + v_s)v + (\alpha + v)\beta\kappa - \frac{k+1}{t}(\alpha + v), \\ \rho_t + (\rho v)_s &= -\rho \frac{C_{ts} \cdot T}{(\alpha_s - \beta\kappa)} \quad (\text{from (3.11)}).\end{aligned}$$

We can now rewrite the system as follows,

$$\begin{aligned}\alpha_t &= \underbrace{C_{tt} \cdot T}_{\text{free}} + \beta(\beta_s + \alpha\kappa), \\ \beta_t &= -\beta_s v - (\alpha + v)\beta_s - (\alpha + v)^2 \kappa - \frac{k+1}{t} \beta + \lambda k^2 t^{k-2} \frac{f}{\rho}, \\ (\alpha + v)_t &= -(\alpha + v)_s v + \beta\beta_s + (\alpha + v)\beta\kappa - \frac{k+1}{t}(\alpha + v), \\ \rho_t &= -\rho_s v - \rho(\alpha + v)_s + \rho\beta\kappa,\end{aligned}$$

where we can see that freedom to choose $C_{tt} \cdot T$ is equivalent to freedom to choose the evolution of α . As such, we may conveniently choose $\alpha_t = 0$. Assuming that we start out with zero initial velocity ($\alpha = \beta = 0$) this would mean α remains zero, yielding the following simplified system:

$$\begin{aligned}\beta_t + v\beta_s &= -v(\beta_s + v\kappa) - \frac{k+1}{t} \beta + \lambda k^2 t^{k-2} \frac{f}{\rho}, \\ v_t + vv_s &= \beta(\beta_s + v\kappa) - \frac{k+1}{t} v, \\ \rho_t + v\rho_s &= \rho(\beta\kappa - v_s).\end{aligned}$$

Finally, we may transform the system by defining $V = vT + \beta N$ to avoid the explicit calculation

of curvature. Noting that

$$V_s = (vT + \beta N)_s = (v_s - \beta\kappa)T + (\beta_s + v\kappa)N$$

and, substituting $\alpha = 0$ into (3.3), to obtain

$$T_t = \beta_s N \quad \text{and} \quad N_t = -\beta_s T,$$

we may compute

$$\begin{aligned} V_t &= (vT + \beta N)_t = (v_t - \beta\beta_s)T + (\beta_t + v\beta_s)N \\ &= \left(\underbrace{\beta(\beta_s + v\kappa) - vv_s - \frac{k+1}{t}v - \beta\beta_s}_{v_t} \right) T \\ &\quad + \left(\underbrace{-v(\beta_s + v\kappa) - v\beta_s - \frac{k+1}{t}\beta + \lambda k^2 t^{k-2} \frac{f}{\rho} + v\beta_s}_{\beta_t} \right) N \\ &= -v \left(\underbrace{(v_s - \beta\kappa)T + (\beta_s + v\kappa)N}_{V_s} \right) - \frac{k+1}{t} \underbrace{(vT + \beta N)}_V + \left(\lambda k^2 t^{k-2} \frac{f}{\rho} \right) N \end{aligned}$$

as well as

$$\rho_t + \underbrace{v}_{V \cdot T} \rho_s = \rho \underbrace{(\beta\kappa - v_s)}_{V_s \cdot T}.$$

REFERENCES

- [1] S. ANGENENT, S. HAKER, AND A. TANNENBAUM, *Minimizing flows for the Monge-Kantorovich problem*, SIAM J. Math. Anal., 35 (2003), pp. 61–97.
- [2] E. BARDELLI, M. COLOMBO, A. MENNUCCI, AND A. YEZZI, *Multiple object tracking via prediction and filtering with a Sobolev-type metric on curves*, in European Conference on Computer Vision, Springer, Berlin, 2012, pp. 143–152.
- [3] J. BENAMOU AND Y. BRENIER, *A computational fluid mechanics solution to the Monge-Kantorovich mass transfer problem*, Numer. Math., 84 (2000), pp. 375–393.
- [4] M. BENYAMIN, J. CALDER, G. SUNDARAMOORTHY, AND A. J. YEZZI, *Accelerated variational PDE's for efficient solution of regularized inversion problems*, J. Math. Imaging Vision, 62 (2020), pp. 10–36, <https://doi.org/10.1007/s10851-019-00910-2>.
- [5] S. BOYD AND L. VANDENBERGHE, *Convex Optimization*, Cambridge University Press, Cambridge, 2004.
- [6] S. BUBECK, Y. T. LEE, AND M. SINGH, *A Geometric Alternative to Nesterov's Accelerated Gradient Descent*, preprint, arXiv:1506.08187 (2015).
- [7] V. CASELLES, R. KIMMEL, AND G. SAPIRO, *Geodesic active contours*, Int. J. Comput. Vis., 22 (1997), pp. 61–79.
- [8] A. CHAMBOLLE AND T. POCK, *A first-order primal-dual algorithm for convex problems with applications to imaging*, J. Math. Imaging Vision, 40 (2011), pp. 120–145.
- [9] T. CHAN AND L. VESE, *Active contours without edges*, IEEE Trans. Image Process., 10 (2001), pp. 266–277.
- [10] T. F. CHAN, S. ESEDOGLU, AND M. NIKOLOVA, *Algorithms for finding global minimizers of image segmentation and denoising models*, SIAM J. Appl. Math., 66 (2006), pp. 1632–1648.

- [11] G. CHARPIAT, O. FAUGERAS, AND R. KERIVEN, *Approximations of shape metrics and application to shape warping and empirical shape statistics*, *Found. Comput. Math.*, 5 (2005), pp. 1–58.
- [12] G. CHARPIAT, R. KERIVEN, J. PONS, AND O. FAUGERAS, *Designing spatially coherent minimizing flows for variational problems based on active contours*, in *International Conference on Computer Vision*, IEEE Computer Society, Los Alamitos, CA, pp. 1403–1408, 2005.
- [13] P. CHAUDHARI, A. OBERMAN, S. OSHER, S. SOATTO, AND G. CARLIER, *Deep relaxation: Partial differential equations for optimizing deep neural networks*, *Res. Math. Sci.*, 5 (2018), 30.
- [14] P. CHAUDHARI AND S. SOATTO, *Stochastic Gradient Descent Performs Variational Inference, Converges to Limit Cycles for Deep Networks*, 2018 Information Theory and Applications Workshop (ITA), San Diego, CA, 2018, pp. 1–10, <https://doi.org/10.1109/ITA.2018.8503224>.
- [15] N. FLAMMARION AND F. BACH, *From averaging to acceleration, there is only a step-size*, *Proc. Mach. Learn. Res.*, 40, 2015, pp. 658–695.
- [16] S. GHADIMI AND G. LAN, *Accelerated gradient methods for nonconvex nonlinear and stochastic programming*, *Math. Program.*, 156 (2016), pp. 59–99.
- [17] H. GOLDSTEIN, C. P. POOLE, AND J. L. SAFKO, *Classical Mechanics*, Addison-Wesley, San Francisco, 2002.
- [18] T. GOLDSTEIN, X. BRESSON, AND S. OSHER, *Geometric applications of the split Bregman method: Segmentation and surface reconstruction*, *J. Sci. Comput.*, 45 (2010), pp. 272–293.
- [19] B. K. HORN AND B. G. SCHUNCK, *Determining optical flow*, *Arti. Intel.*, 17 (1981), pp. 185–203.
- [20] C. HU, W. PAN, AND J. T. KWOK, *Accelerated gradient methods for stochastic optimization and online learning*, in *Advances in Neural Information Processing Systems 22*, Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams, and A. Culotta, eds., Curran Associates, Red Hook, NY, 2009, pp. 781–789.
- [21] S. JI AND J. YE, *An accelerated gradient method for trace norm minimization*, in *Proceedings of the 26th Annual International Conference on Machine Learning, ICML '09*, ACM, New York, 2009, pp. 457–464.
- [22] V. JOJIC, S. GOULD, AND D. KOLLER, *Accelerated dual decomposition for map inference*, in *Proceedings of the 27th International Conference on International Conference on Machine Learning, ICML'10*, ACM, Red Hook, New York, 2010, pp. 503–510.
- [23] L. KANTOROVICH, *On the transfer of masses*, *Dokl. Akad. Nauk. SSSR*, 37 (1942), pp. 227–220.
- [24] L. KANTOROVICH, *On a problem of Monge*, *Uspekhi Mat. Nauk.*, 3 (1948), pp. 225–226.
- [25] S. KICHENASSAMY, A. KUMAR, P. OLVER, A. TANNENBAUM, AND A. YEZZI, *Conformal curvature flows: From phase transitions to active vision*, *Arch. Ration. Mech. Anal.*, 134 (1996), pp. 275–301.
- [26] K. KOLEV, T. BROX, AND D. CREMERS, *Fast joint estimation of silhouettes and dense 3d geometry from multiple images*, *IEEE Trans. Pattern Anal. Mach. Intell.*, 34 (2012), pp. 493–505.
- [27] W. KRICHENE, A. BAYEN, AND P. L. BARTLETT, *Accelerated mirror descent in continuous and discrete time*, in *Advances in Neural Information Processing Systems 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, eds., Curran Associates, Red Hook, NY, 2015, pp. 2845–2853.
- [28] H. LI AND Z. LIN, *Accelerated proximal gradient methods for nonconvex programming*, in *Advances in Neural Information Processing Systems 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, eds., Curran Associates, Red Hook, NY, 2015, pp. 379–387.
- [29] A. MENNUCCI, A. YEZZI, AND G. SUNDARAMOORTHY, *Properties of Sobolev-type metrics in the space of curves*, *Interfaces Free Boundaries*, 10 (2008), pp. 423–445.
- [30] G. MONGE, *Mémoire sur la théorie des déblais et des remblais*, Imprimerie Royale, Paris, 1781.
- [31] I. MUKHERJEE, K. CANINI, R. FRONGILLO, AND Y. SINGER, *Parallel boosting with momentum*, in *Machine Learning and Knowledge Discovery in Databases*, H. Blockeel, K. Kersting, S. Nijssen, and F. Zelezny, eds., Springer, Berlin, 2013, pp. 17–32.
- [32] Y. NESTEROV, *A method of solving a convex programming problem with convergence rate $o(1/k^2)$* , in *Sov. Math. Dokl.*, 27 (1983), pp. 372–376.
- [33] Y. NESTEROV, *Smooth minimization of non-smooth functions*, *Math. Program.*, 103 (2005), pp. 127–152.
- [34] Y. NESTEROV, *Accelerating the cubic regularization of Newton's method on convex problems*, *Math. Program.*, 112 (2008), pp. 159–181.
- [35] Y. NESTEROV, *Gradient methods for minimizing composite functions*, *Math. Program.*, 140 (2013), pp. 125–161.

- [36] Y. NESTEROV, *Introductory Lectures on Convex Optimization: A Basic Course*, 1st ed., Kluwer, Boston, 2014.
- [37] Y. NESTEROV AND B. T. POLYAK, *Cubic regularization of Newton method and its global performance*, Math. Program., 108 (2006), pp. 177–205.
- [38] M. NIETHAMER AND A. TANNENBAUM, *Dynamic level sets for visual tracking*, in Proceedings of IEEE Conference on Decision and Control, IEEE, Piscataway, NJ, 2003, pp. 4883–4888.
- [39] M. NIETHAMER AND A. TANNENBAUM, *Dynamic geodesic snakes for visual tracking*, in IEEE Conference on Computer Vision and Pattern Recognition, IEEE Computer Society, Los Alamitos, CA, 2004, pp. 4883–4888.
- [40] M. NIETHAMER AND A. TANNENBAUM, *Dynamic geodesic snakes for visual tracking*, IEEE Trans. Automat. Control, 51 (2006), pp. 562–579.
- [41] M. NIETHAMER, P. VELA, AND A. TANNENBAUM, *Geometric observers for dynamically evolving curves*, IEEE Trans. Pattern Anal. Mach. Intell., 30 (2008), pp. 1093–1108.
- [42] B. O'DONOGHUE AND E. CANDÈS, *Adaptive restart for accelerated gradient schemes*, Found. Comput. Mech., 15 (2015), pp. 715–732.
- [43] S. OSHER AND N. PARAGIOS, *Geometric Level Set Methods in Imaging, Vision and Graphics*, Springer, New York, 2003.
- [44] S. OSHER AND J. SETHIAN, *Fronts propagation with curvature dependent speed: Algorithms based on Hamilton-Jacobi formulations*, J. Comput. Phys., 79 (1988), pp. 12–49.
- [45] T. POCK, A. CHAMBOLLE, D. CREMERS, AND H. BISCHOF, *A convex relaxation approach for computing minimal partitions*, in Computer Vision and Pattern Recognition, IEEE Computer Society, Los Alamitos, CA, 2009, pp. 810–817.
- [46] G. SAPIRO, *Geometric Partial Differential Equations and Image Analysis*, Cambridge Press, Cambridge, 2000.
- [47] J. SETHIAN, *Level Set Methods: Evolving Interfaces in Geometry, Fluid Mechanics, Computer Vision, and Material Science*, Cambridge University Press, Cambridge, 1996.
- [48] W. SU, S. BOYD, AND E. CANDÈS, *A differential equation for modeling Nesterov's accelerated gradient method: Theory and insights*, in Advances in Neural Information Processing Systems, Curran Associates, Red Hook, NY, 2014, pp. 2510–2518.
- [49] G. SUNDARAMOORTHY, J. JACKSON, A. YEZZI, AND A. MENNUCCI, *Tracking with Sobolev active contours*, in IEEE Conference on Computer Vision and Pattern Recognition, IEEE Computer Society, Los Alamitos, CA, 2006, pp. 674–680.
- [50] G. SUNDARAMOORTHY, A. MENNUCCI, S. SOATTO, AND A. YEZZI, *A new geometric metric in the space of curves, and applications to tracking deforming objects by prediction and filtering*, SIAM J. Imaging Sci., 4 (2011), pp. 109–145.
- [51] G. SUNDARAMOORTHY, A. MENNUCCI, A. YEZZI, AND S. SOATTO, *Tracking deforming objects by filtering and prediction in the space of curves*, in Proceedings of the IEEE Conference on Decision and Control, Piscataway, NJ, 2009, pp. 2395–2401.
- [52] G. SUNDARAMOORTHY AND A. YEZZI, *Variational PDEs for acceleration on manifolds and application to diffeomorphisms*, in Advances in Neural Information Processing Systems 31, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, eds., Curran Associates, Red Hook, NY, 2018, pp. 3793–3803, <http://papers.nips.cc/paper/7636-variational-pdes-for-acceleration-on-manifolds-and-application-to-diffeomorphisms.pdf>.
- [53] G. SUNDARAMOORTHY, A. YEZZI, AND A. MENNUCCI, *Sobolev active contours*, in Workshop on Variational Geometric and Level Set Methods in Computer Vision, Springer, Berlin, 2005, pp. 109–120.
- [54] G. SUNDARAMOORTHY, A. YEZZI, AND A. MENNUCCI, *Sobolev active contours*, Int. J. Comput. Vis., 7 (2007), pp. 345–366.
- [55] G. SUNDARAMOORTHY, A. YEZZI, AND A. MENNUCCI, *Coarse-to-fine segmentation and tracking using Sobolev active contours*, IEEE Trans. Pattern Anal. Mach. Intell., 30 (2008), pp. 851–864.
- [56] G. SUNDARAMOORTHY AND A. J. YEZZI, *Accelerated Optimization in the PDE Framework: Formulations for the Manifold of Diffeomorphisms*, preprint, <https://arxiv.org/abs/1804.02307> (2018).
- [57] J. L. TROUTMAN, *Variational Calculus and Optimal Control*, Springer, New York, 1996.
- [58] C. VILLANI, *Topics in Optimal Transportation*, Grad. Stud. Math. 58, AMS, Providence RI, 2003.

- [59] A. WIBISONO, A. C. WILSON, AND M. I. JORDAN, *A variational perspective on accelerated methods in optimization*, Proc. Natl. Acad. Sci. USA, 113 (2016), 201614734.
- [60] Y. YANG AND G. SUNDARAMOORTHY, *Shape tracking with occlusions via coarse-to-fine region based Sobolev descent*, IEEE Trans. Pattern Anal. Mach. Intell., 37 (2015), pp. 1053–1066.
- [61] A. YEZZI AND S. SOATTO, *Stereoscopic segmentation*, Int. J. Comput. Vis., 53 (2003), pp. 31–43.
- [62] A. YEZZI, G. SUNDARAMOORTHY, AND M. BENYAMIN, *PDE acceleration for active contours*, in Proceedings of The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE Computer Society, Los Alamitos, CA, 2019, pp. 12310–12320.
- [63] A. J. YEZZI AND S. SOATTO, *Stereoscopic segmentation*, in Proceedings of the Eighth IEEE International Conference on Computer Vision. ICCV 2001, Vol. 1, IEEE Computer Society, Los Alamitos, CA, 2001, pp. 59–66.
- [64] A. J. YEZZI AND G. SUNDARAMOORTHY, *Accelerated Optimization in the PDE Framework: Formulations for the Active Contour Case*, preprint, <https://arxiv.org/abs/1711.09867> (2017).