



# Accelerated Variational PDEs for Efficient Solution of Regularized Inversion Problems

Minas Benyamin<sup>1</sup> · Jeff Calder<sup>2</sup> · Ganesh Sundaramoorthi<sup>3</sup> · Anthony Yezzi<sup>1</sup>

Received: 2 April 2019 / Accepted: 14 September 2019 / Published online: 30 September 2019  
© Springer Science+Business Media, LLC, part of Springer Nature 2019

## Abstract

We further develop a new framework, called *PDE acceleration*, by applying it to calculus of variation problems defined for general functions on  $\mathbb{R}^n$ , obtaining efficient numerical algorithms to solve the resulting class of optimization problems based on simple discretizations of their corresponding accelerated PDEs. While the resulting family of PDEs and numerical schemes are quite general, we give special attention to their application for regularized inversion problems, with particular illustrative examples on some popular image processing applications. The method is a generalization of momentum, or accelerated, gradient descent to the PDE setting. For elliptic problems, the descent equations are a nonlinear damped wave equation, instead of a diffusion equation, and the acceleration is realized as an improvement in the CFL condition from  $\Delta t \sim \Delta x^2$  (for diffusion) to  $\Delta t \sim \Delta x$  (for wave equations). We work out several explicit as well as a semi-implicit numerical scheme, together with their necessary stability constraints, and include recursive update formulations which allow minimal-effort adaptation of existing gradient descent PDE codes into the accelerated PDE framework. We explore these schemes more carefully for a broad class of regularized inversion applications, with special attention to quadratic, Beltrami, and total variation regularization, where the accelerated PDE takes the form of a nonlinear wave equation. Experimental examples demonstrate the application of these schemes for image denoising, deblurring, and inpainting, including comparisons against primal–dual, split Bregman, and ADMM algorithms.

**Keywords** Nesterov acceleration · Accelerated gradient descent · PDE acceleration · Nonlinear wave equations · Image denoising · Image deblurring · Image restoration · Total variation · Beltrami regularization

**Mathematics Subject Classification** 97N40 · 65M06 · 35Q93 · 65K10

---

J. Calder was supported by NSF–DMS Grant 1713691, and A. Yezzi was supported by NSF–CCF Grant 1526848 and ARO W911NF–18–1–0281, and NIH R01 HL143350.

---

✉ Jeff Calder  
jcalder@umn.edu

Minas Benyamin  
minasbenyamin@gatech.edu

Ganesh Sundaramoorthi  
ganesh.sun@gmail.com

Anthony Yezzi  
anthony.yezzi@ece.gatech.edu

<sup>1</sup> School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, Georgia

<sup>2</sup> School of Mathematics, University of Minnesota, Minneapolis, USA

<sup>3</sup> United Technologies Research Center, East Hartford, CT, USA

## 1 Introduction

Variational problems have found great success and are widely used, in image processing for problems such as noisy or blurry image restoration, image inpainting, image decomposition, and many other problems [2]. Many image processing problems have the form

$$\min_u \int_{\Omega} L(x, u, \nabla u) \, dx \quad (1)$$

where  $L$  is convex in  $\nabla u$ <sup>1</sup> and the corresponding gradient descent equation

---

<sup>1</sup> Nonconvex problems are also widely used, see, e.g., [17].

$$u_t + L_z(x, u, \nabla u) - \operatorname{div}(\nabla_p L(x, u, \nabla u)) = 0$$

is a nonlinear diffusion equation, where  $L = L(x, z, p)$ . Solving (1), an elliptic problem, via gradient descent is inefficient, due in large part to the stiff stability (CFL) condition  $\Delta t \leq C \Delta x^2$  for diffusion equations. This has led to the development of more efficient optimization algorithms, such as primal–dual methods [9] and the split Bregman approach [12] that avoid this numerical stiffness.<sup>2</sup>

Optimization is also widely used in machine learning. Typically for both modern machine learning and image processing problems, first-order methods based on computing only the gradient are preferable, since computing and storing the Hessian are intractable [4]. Although this generalization is with the caveat, most types of machine learning optimization problems are often structurally different than in image processing.

For both machine learning and image processing, discrete gradient descent is typically written as

$$x_{k+1} = x_k - \alpha \nabla f(x_k), \quad (2)$$

where in machine learning the time step  $\alpha$  is called the *learning rate*. And although gradient descent is provably convergent for convex problems [5], the method can be very slow to converge in practice.

To address this issue, many versions of *accelerated* gradient descent, typically described as momentum-based techniques, have been proposed in the literature and are widely used in machine learning [29]. At some heuristic level, gradient descent is often slow to converge because the local descent direction is not reliable on a larger scale, leading to large steps in poor directions and large corrections in the opposite direction. The descent is also dependent on the magnitude of the gradient, which slows or even traps descent when it is locally flat.

Accelerated descent methods typically incorporate some type of averaging of past descent directions, which provides a superior descent direction compared to the local gradient. One of the oldest accelerated methods is Polyak’s heavy ball method [18]

$$x_{k+1} = x_k - \alpha \nabla f(x_k) + \beta(x_k - x_{k-1}). \quad (3)$$

The term  $\beta(x_k - x_{k-1})$  acts to average the local descent direction with the previous direction and is referred to as *momentum*. Polyak’s heavy ball method was studied in the continuum by Attouch et al. [1] and also by Goudou and Munier [13], who call it the *heavy ball with friction*. In the

<sup>2</sup> Primal–dual and split Bregman also avoid the non-smoothness of the  $L^1$  norm, which is an issue in descent-based approaches and often requires some form of regularization.

continuum, Polyak’s heavy ball method corresponds to the equations of motion for a body in a potential field, which is the second-order ODE

$$\ddot{x} + a\dot{x} = -\nabla f(x). \quad (4)$$

A more recent example of momentum descent is the famous Nesterov accelerated gradient descent [16]

$$\begin{aligned} x_{k+1} &= y_k - \alpha \nabla f(y_k) \\ y_{k+1} &= x_{k+1} + \frac{k-1}{k+2}(x_{k+1} - x_k). \end{aligned} \quad (5)$$

In [16], Nesterov proved a convergence rate of  $O(1/k^2)$  after  $k$  iterations for smooth convex problems. This is provably optimal for first-order methods.

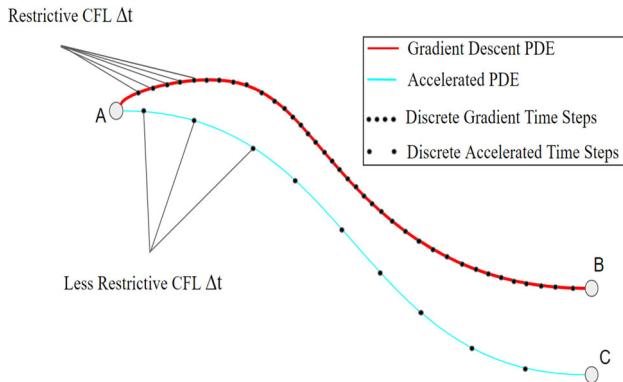
The seminal works of Polyak and Nesterov have spawned a whole field of momentum-based descent methods, and variants of these methods are widely used in machine learning, such as the training of neural networks in deep learning [25,29]. The methods are popular for both their superior convergence rates for convex problems, but also their ability to avoid local minima in nonconvex problems, which is not fully understood in a rigorous sense. There has been significant interest recently in understanding the Nesterov accelerated descent methods. In particular, Su et al. [22] recently showed that Nesterov acceleration is simply a discretization of the second-order ODE

$$\ddot{x} + \frac{3}{t}\dot{x} = -\nabla f(x). \quad (6)$$

Other works have since termed this ODE as *continuous time Nesterov* [27]. We note the friction coefficient  $3/t$  vanishes as  $t \rightarrow \infty$ , which explains why many implementations of Nesterov acceleration involve *restarting*, or resetting the time to  $t = 0$  when the system is underdamped [27].

However, it is the work of Wibisono et al. [29] that gives the clearest picture of Nesterov acceleration. They show that virtually all Nesterov accelerated gradient descent methods are simply discretizations of the ODE equations of motion for a particular Lagrangian action functional. This endows Nesterov acceleration with a variational framework, which aids in our understanding, and more importantly can be easily adapted to other settings. This framework was extended to the partial differential equation (PDE) setting by Sundaramoorthy and Yezzi in their initial works [23,24,30,31] where the first set of *accelerated PDEs* were formulated both for geometric flows of contours and surfaces (active contours) as well as for diffeomorphic mappings between images (optical flow).

In this manuscript, acceleration is addressed both in the continuum (space and time) and for a class of discrete problems. Our primary intended usage of the term “acceleration”



**Fig. 1** Illustration of gradient descent PDE vs accelerated gradient PDE: gradient descent PDEs and the corresponding accelerated gradient PDEs generally follow different paths. Further, accelerated PDEs lead to discretization schemes with less restrictive CFL conditions corresponding to larger discrete steps than gradient descent PDEs, leading to faster convergence in the convex case. Note in the case of strictly convex problems  $B = C$

refers to the physical interpretation of acceleration as the second time derivative of the evolving entity. In typical gradient descent, the gradient defines the velocity of the evolution. In accelerated descent, the gradient defines the acceleration of the evolution. Often, the gradient descent PDE for an energy function  $E(u)$  has the form

$$u_t = -\nabla E(u), \quad (7)$$

whereas what we are calling an accelerated gradient PDE generally has the form

$$u_{tt} = -au_t - \nabla E(u), \quad (8)$$

where the addition of  $au_t$ , with damping coefficient  $a$ , acts as a friction term to dissipate kinetic energy. Our secondary usage of the term acceleration applies to a special class of variational problems where a less restrictive CFL condition allows coarser time sampling of the discretized PDEs. Here an improvement in the CFL conditions for these PDEs is realized numerically by much larger step sizes for both their explicit and semi-implicit discretizations. Thereby, this allows another interpretation of the term “acceleration” as a raw computational speedup, as shown conceptually in Fig. 1.

We note there are also some acceleration-type methods that have appeared recently in image processing [3, 6, 14, 19, 28], with the exception of [6, 19], these methods are not derived from a variational framework, and so they lack energy monotonicity and convergence guarantees. Ratner and Zeevi [19] do not derive their approach from an action integral and therefore do not have the connections to Nesterov or the heavy ball method. They do not address stability or model analysis and likewise lack the convergence guarantees. Baravdish, Svensson, Gulliksson and Zhang do, however,

address the fully linear case of the same nonlinear setup, motivated from the heavy ball method, but restricted to the  $L^p$  norm of gradient as a regularizer, focusing on mainly  $p = 1$  (total variation) experiments. They prove the existence/uniqueness of weak solutions for a regularized version of the damped nonlinear wave equation and prove the exponential convergence rate from [8]. They, however, limit their analysis to a single discretization scheme, the Stormer–Verlet method, which is different from ours. Furthermore, they do not derive their approach from an action integral and thus lack connections to Nesterov. Our paper extends the analysis to multiple discretization schemes and multiple fidelities and gives a more thorough discussion of the stability (CFL) conditions for the PDEs. We also extend the connection of Nesterov acceleration to semi-implicit Euler.

Acceleration has also been applied to composite functionals, which have a strongly convex smooth term and a non-smooth convex term [7, 10]. These methods are based on accelerating the forward/backward splitting algorithms for such composite functionals and use a regularization on the TV seminorm. We show in this paper that PDE acceleration can be applied directly to the non-smooth unregularized TV seminorm, due to a nonlinear stability condition we discuss in Sect. 4.5.

## 1.1 Contributions

The contributions of this paper are:

- 1) We extend the class of *accelerated PDEs* formulated in [23, 24, 30, 31] to the setting of generic functions over  $\mathbb{R}^n$ , building on the variational insights pioneered by [29]. The method applies to solving general problems in the calculus of variations. In similar spirit to (4) and (6), the descent equations in PDE acceleration correspond to a continuous second-order flow in time which, for a broad class of regularized inversion problems to be addressed in Sect. 4, take on the specific form of damped nonlinear wave equations rather than the reaction–diffusion equations that arise as their traditional gradient descent counterparts. Accelerated PDEs can be solved numerically with simple explicit Euler or semi-implicit Euler schemes which we develop in Sect. 3.
- 2) We realize an improvement in the CFL condition from  $\Delta t \sim \Delta x^2$  for diffusion equations (or standard gradient descent), to  $\Delta t \sim \Delta x$  for wave equations. We also refer to this improved stability condition as acceleration since the maximum stable time step that the PDEs can take has increased.<sup>3</sup> In fact, we will show early on in Sect. 3 that the improvements in the CFL condition for

<sup>3</sup> Acceleration can also be used in the sense that in Nesterov acceleration the gradient is forward looking and computed ahead of the current

explicit numerical accelerated PDE schemes (compared with their gradient descent counterparts) are a completely general property of accelerated PDEs which applies even when the wave equation structure does not arise.

3) We apply the new numerical discretizations to a class of inversion problems: denoising, deblurring, and inpainting with various regularizers and show robust performance and improvements over existing methods.

In a companion paper [8], we study the PDE acceleration method rigorously and prove a convergence rate, perform a complexity analysis, and show how to optimally select the parameters, including the damping coefficient (these results are summarized in Sect. 2). That paper, however, does not analyze CFL conditions or stability, is more focused on convergence rate analysis, and addresses applications to minimal surface obstacle problems and improvements to a primal–dual method.

## 1.2 Paper Outline

This manuscript is organized as follows: In Sect. 2, we summarize the work of a companion paper [8], in which we rigorously study the PDE acceleration method, prove a convergence rate, perform a complexity analysis, and show how to optimally select the tuning parameters including the damping coefficient. In Sect. 3, we develop the explicit and semi-implicit Euler schemes and derive the improved CFL conditions for several explicit and semi-implicit numerical accelerated PDE schemes. In Sect. 4, we extend the derived numerical schemes: first-order accelerated, second-order accelerated, and semi-implicit scheme to two separate inversion problems for both Beltrami and TV regularization. In Sect. 5, we apply the method to quadratic, Beltrami, and total variation regularized problems in image processing including denoising, deblurring, and inpainting, obtaining results that are comparable to state-of-the-art methods, such as the split Bregman approach, and ADMM, and superior to primal–dual methods.

## 2 PDE Acceleration

We now present our PDE acceleration framework, which is based on the seminal work of [23,24,29–31] with suitable modifications to image processing problems. We consider the calculus of variation problem

$$\min_u E[u] := \int_{\Omega} \Phi(x, \nabla u) + \Psi(x, u) \, dx.$$

state [16]. The semi-implicit case which is an extension from the ODE framework of Nesterov uses a similar look ahead for its update scheme.

The Euler–Lagrange equation satisfied by minimizers is

$$\nabla E[u] := \Psi_z(x, u) - \operatorname{div}(\nabla \Phi(x, \nabla u)) = 0, \quad (9)$$

where  $\Phi = \Phi(x, p)$ ,  $\nabla \Phi = \nabla_p \Phi$ , and  $\Psi = \Psi(x, z)$ . We note that the gradient  $\nabla E[u]$  satisfies

$$\frac{d}{d\varepsilon} \Big|_{\varepsilon=0} E[u + \varepsilon v] = \int_{\Omega} \nabla E[u] v \, dx \quad (10)$$

for all  $v$  smooth with compact support and is often called the  $L^2$ -gradient due to the presence of the  $L^2$  inner product on the right-hand side.

We define the action integral

$$J[u] = \int_{t_0}^{t_1} k(t) \left( \frac{1}{2} \int_{\Omega} \rho u_t^2 \, dx - b(t) E[u] \right) dt, \quad (11)$$

where  $k(t)$  and  $b(t)$  are time-dependent weights,  $\rho = \rho(x)$  represents a mass density, and  $u = u(x, t)$ . Notice the action integral is the weighted difference between kinetic energy  $\frac{1}{2} \int \rho u_t^2 \, dx$  and potential energy  $E[u]$ . The PDE accelerated descent equations are defined to be the equations of motion in the Lagrangian sense corresponding to the action  $J$ . To compute the equations of motion, we take a variation on  $J$  to obtain

$$\begin{aligned} 0 &= \frac{d}{d\varepsilon} \Big|_{\varepsilon=0} J[u + \varepsilon v] \\ &= \int_{t_0}^{t_1} \int_{\Omega} k(t) \rho u_t v_t - k(t) b(t) \nabla E[u] v \, dx, \end{aligned}$$

for smooth  $v$  with compact support in  $\Omega \times (t_0, t_1)$ . Integrating by parts in  $t$ , we have

$$0 = \int_{t_0}^{t_1} \int_{\Omega} \left[ -\frac{\partial}{\partial t} (k(t) \rho u_t) - k(t) b(t) \nabla E[u] \right] v \, dx.$$

Thus, the PDE accelerated descent equations are

$$\frac{\partial}{\partial t} (k(t) \rho u_t) = -k(t) b(t) \nabla E[u].$$

It is more convenient to define  $a(t) = k'(t)/k(t)$  and rewrite the descent equations as

$$u_{tt} + a(t) u_t = -b(t) \rho(x)^{-1} \nabla E[u]. \quad (12)$$

For image processing problems, there is typically no Dirichlet boundary condition, so the natural variational boundary condition  $\nabla_p \Phi(x, \nabla u) \cdot \mathbf{n} = 0$  is imposed on the boundary  $\partial\Omega$ , where  $\mathbf{n}$  is the outward normal. Often this reduces to the Neumann condition  $\frac{\partial u}{\partial \mathbf{n}} = 0$ .

In a companion paper [8], we study PDE acceleration descent Eq. (12) rigorously. In particular, we prove energy

monotonicity and a linear convergence rate. We summarize the results in Lemma 1 and Theorem 1.

**Lemma 1** (Energy monotonicity [8]) *Assume  $a(t), b(t) \geq 0$  and let  $u$  satisfy (12). Suppose either  $u(x, t) = g(x)$  or  $\nabla \Phi(x, \nabla u) \cdot \mathbf{n} = 0$  on  $\partial\Omega$ . Then*

$$\frac{d}{dt} (K[u] + b(t)E[u]) = -2a(t)K[u] + b'(t)E[u], \quad (13)$$

where  $K[u] = \frac{1}{2} \int_{\Omega} \rho u_t^2 dx$ . In particular, total energy is always decreasing provided  $b'(t) \leq 0$  and  $E[u] \geq 0$ .

**Theorem 1** (Convergence rate [8]) *Let  $u$  satisfy (12) and let  $u^*$  be a solution of  $\nabla E[u^*] = 0$  in  $\Omega$ . Assume  $\Phi$  is uniformly convex in  $\nabla u$ ,  $\Psi$  is convex, and  $\Psi_{zz}$  is bounded above,  $u = u^*$  on  $\partial\Omega$ ,  $a(t) = a > 0$  is constant, and  $b(t) \equiv 1$  and  $\rho \equiv 1$ . Then there exists  $C, \beta > 0$  such that*

$$\|u - u^*\|_{H^1(\Omega)}^2 \leq C \exp(-\beta t). \quad (14)$$

We mention that the same convergence rate (14) holds for gradient descent

$$u_t = -\nabla E[u]$$

under the same conditions on  $E$ . The difference is that gradient descent is a diffusion equation, which requires a time step of  $\Delta t \sim \Delta x^2$  for stability, while PDE acceleration (12) is a wave equation which allows much larger time steps  $\Delta t \sim \Delta x$ . Thus, the acceleration is realized as a relaxation in the CFL condition.

While Theorem 1 provides a convergence rate, it does not give advice on how to select the damping coefficient  $a > 0$ . It was shown in [8] how to optimally select the damping coefficient in the linear setting, and we find this choice is useful for nonlinear problems as well. For convenience, we recall the results from [8], which apply to the linear PDE acceleration equation

$$u_{tt} + au_t + Lu + \lambda u = f \quad \text{in } \Omega \times (0, \infty), \quad (15)$$

where  $L$  is a linear second-order elliptic operator. A Fourier analysis [8] leads to the optimal choice

$$a = 2\sqrt{\lambda_1 + \lambda}, \quad (16)$$

where  $\lambda_1$  is the first Dirichlet eigenvalue of  $L$  (or for the Neumann problem, the first eigenvalue corresponding to a nontrivial eigenfunction), and the optimal convergence rate

$$|u(x, t) - u^*(x)| \leq C \exp(-at). \quad (17)$$

Notice that if  $L$  is degenerate elliptic, so  $\lambda_1 = 0$ , which roughly corresponds to a non-strongly convex optimization

problem, the method still converges when  $\lambda > 0$ , that is, the presence of a fidelity term in the image processing problem enables, and accelerates, convergence. This suggests why the algorithm is successful even for TV restoration, which is not strongly convex but has a fidelity.

### 3 Numerical Schemes for Accelerated PDEs

We now describe various time discretization strategies for the generic accelerated PDE

$$u_{tt} + au_t = -\nabla E[u, u_x, u_{xx}, \dots], \quad (18)$$

alongside related discretizations of the generic gradient descent PDE

$$u_t = -\nabla E[u, u_x, u_{xx}, \dots] \quad (19)$$

for comparison. Note (18) represents the unit density ( $\rho = 1$ ) and unit energy scaled ( $b = 1$ ) case of (12). A key advantage of accelerated PDE schemes for regularized inversion problems, which we explore subsequently in Sect. 4, is that in typical cases where the gradient descent PDE (19) takes the form of a linear or nonlinear reaction-diffusion equation, the matching accelerated PDE (18) takes the form of a linear or nonlinear wave equation, whose explicit time discretization permits a much larger stable time step than the explicit discretization of (19). Therefore, due to their simplicity of implementation, as well as their immediately parallelizable structure, we will restrict our discussion to explicit update schemes and to the semi-implicit Euler scheme whose two-part update consists of partial updates which are both explicit in nature.

#### 3.1 Explicit Forward Euler for Gradient Descent PDEs

We start by considering the explicit forward Euler discretization of the continuous gradient descent PDE (19). Using a forward difference in time to approximate the time derivative on the left-hand side, we obtain

$$\frac{u(x, t + \Delta t) - u(x, t)}{\Delta t} = -\nabla E.$$

This leads to the following simple discrete iteration

$$\begin{aligned} \Delta u^n(x) &= -\Delta t \nabla E^n \\ u^{n+1}(x) &= u^n(x) + \Delta u^n(x) \end{aligned} \quad (20)$$

where  $u^n(x) \doteq u(x, n\Delta t)$  denotes the current iterate,  $\Delta u^n \doteq u(x, n\Delta t + \Delta t) - u(x, n\Delta t)$  the increment to be applied,  $u^{n+1}(x) \doteq u(x, (n+1)\Delta t)$  the new iterate, and  $\nabla E^n(x) \doteq$

$\nabla E(x, n\Delta t)$  the discrete approximation of the gradient computed at step  $n$ .

In most cases, stability considerations require an upper bound on the time step  $\Delta t$  (the CFL condition) dependent upon the discretization of  $\nabla E^n$  [26]. Often this upper bound for stable time steps is computed using Von Neumann analysis by linearizing  $\nabla E^n$  in (20) and taking a discrete Fourier transform (DFT) on both sides of the homogeneous part to obtain

$$U^{n+1}(\omega) - U^n(\omega) = -\Delta t [z(\omega)U^n(\omega)].$$

Such a structure often arises when  $\nabla E^n$  is computed explicitly using only the values of  $u^n$ . In such cases, its linearization will consist of a combination of  $u^n$  values whose DFT can be written in the form  $z(\omega)U^n(\omega)$  where  $U^n(\omega)$  denotes the DFT of  $u^n$ . We will refer to  $z(\omega)$  as the *gradient amplifier*,<sup>4</sup> which is defined as

$$\text{gradient amplifier } z(\omega) \doteq \frac{\text{DFT}(L^n)}{\text{DFT}(u^n)}, \quad (21)$$

where  $L^n$  is the linearized homogeneous part of  $\nabla E^n$ . This leads to the following update

$$U^{n+1}(\omega) = \underbrace{(1 - \Delta t z(\omega))}_{\xi(\omega)} U^n(\omega),$$

which will be stable as long as the overall update amplification factor  $\xi(\omega)$  does not have complex amplitude exceeding unity for any frequency  $\omega$ . This condition can be expressed as

$$\xi(\omega)\xi^*(\omega) = (1 - \Delta t z(\omega))(1 - \Delta t z^*(\omega)) \leq 1,$$

which leads to the time step restriction

$$\Delta t \leq \frac{z(\omega) + z^*(\omega)}{z(\omega)z^*(\omega)} = \frac{1}{z(\omega)} + \frac{1}{z^*(\omega)} = 2\Re\left(\frac{1}{z(\omega)}\right).$$

For elliptic operators, which are common in regularized optimization in image processing, the gradient amplifier is real and nonnegative:  $z(\omega) \geq 0$ . In such cases, the stability constraint takes the form of the following CFL condition

$$\Delta t \leq \frac{2}{z_{\max}} \quad (22)$$

where  $z_{\max} \doteq \max_{\omega} z(\omega)$ .

<sup>4</sup> A discrete version of what is often called the *symbol* of the underlying linear differential operator that is being approximated.

### 3.2 Fully Explicit Schemes for Accelerated PDEs

We now turn our attention to the explicit discretizations of the accelerated PDE (18). We will consider both first- and second-order approximations of the time derivatives and will exploit the following lemma in the Von Neumann stability analysis for each of these choices.

**Root Amplitude Lemma** *Given a quadratic equation  $A\xi^2 + B\xi + C = 0$  with real coefficients ( $A \neq 0$ ), its roots will satisfy  $|\xi| \leq 1$  if and only if  $\frac{|B|}{|A|} - 1 \leq \frac{C}{A} \leq 1$  (or equivalently  $A > C$  and  $A + C > |B|$  for positive  $A$ ).*

**Proof** We first prove the result in the special case that  $A = 1$  and  $B \geq 0$ , in which case the roots are  $\xi = -\frac{B}{2} \pm \frac{1}{2}\sqrt{B^2 - 4C}$  and claim that  $|\xi| \leq 1$  if and only if

$$B - 1 \leq C \leq 1.$$

If the roots are imaginary, then both have complex amplitude  $|\xi|^2 = C > 0$  which makes the right hypothesis both necessary and sufficient. The left hypothesis automatically follows since  $C > \frac{B^2}{4} \geq B - 1$  (the first part for the roots to be imaginary and the second part equivalent to  $(B - 2)^2 \geq 0$ ). In the case of real roots, we want the larger magnitude root to satisfy  $|\xi| = \frac{B+1}{2}\sqrt{B^2 - 4C} \leq 1$ , which can be expressed as  $\sqrt{B^2 - 4C} \leq 2 - B$ . This immediately yields  $B \leq 2$  as a necessary condition to keep the right side positive. Under this condition, we can square both sides and simplify to obtain the left hypothesis as necessary and sufficient. The right hypothesis automatically follows since  $C < \frac{B^2}{4} < 1$  (the first part for the roots to be real and the second part based on our condition). Combining the hypotheses yields  $B - 1 \leq 1$  which satisfies the necessary condition, thus completing the special case proof. The general case follows since the roots of  $A\xi^2 + B\xi + C$  have the same magnitude as the roots of  $\xi^2 + \frac{|B|}{|A|}\xi + \frac{C}{A}$ .  $\square$

#### 3.2.1 Second Order in Time Scheme

Using central difference approximations for both time derivatives gives a second-order discretization in time

$$\frac{u(x, t + \Delta t) - 2u(x, t) + u(x, t - \Delta t)}{\Delta t^2} + a \frac{u(x, t + \Delta t) - u(x, t - \Delta t)}{2\Delta t} = -\nabla E(x, t),$$

which leads to the update

$$u^{n+1}(x) = \frac{2u^n - \left(1 - \frac{a\Delta t}{2}\right)u^{n-1} - \Delta t^2 \nabla E^n}{1 + \frac{a\Delta t}{2}}. \quad (23)$$

Applying the DFT to the linearized homogeneous part of the update scheme (23) yields

$$U^{n+1}(\omega) = \frac{(2 - \Delta t^2 z(\omega)) U^n - \left(1 - \frac{a\Delta t}{2}\right) U^{n-1}}{1 + \frac{a\Delta t}{2}}$$

where  $z(\omega)$  denotes the gradient amplifier (21). If we substitute  $U^{n\pm m} = \xi^{\pm m} U^n$ , where  $\xi(\omega)$  denotes the overall update amplification factor, then we obtain the quadratic equation

$$\underbrace{\left(1 + \frac{a\Delta t}{2}\right)}_A \xi^2 + \underbrace{\left(\Delta t^2 z(\omega) - 2\right)}_B \xi + \underbrace{\left(1 - \frac{a\Delta t}{2}\right)}_C = 0.$$

In the case of real  $z(\omega)$ , we may exploit the Root Amplitude Lemma to check the stability condition  $|\xi(\omega)| \leq 1$ . The first condition  $A \geq C$  of the lemma (for positive  $A$ ) is satisfied since  $1 + \frac{a\Delta t}{2} \geq 1 - \frac{a\Delta t}{2}$  for all positive  $a$  and  $\Delta t$ , and so we use the second condition  $A + C \geq |B|$  to obtain the stability condition  $2 \geq |2 - \Delta t^2 z(\omega)|$ , which may be rewritten as  $0 \leq \Delta t^2 z(\omega) \leq 4$ . In the case where  $z(\omega) \geq 0$ , we automatically satisfy the left-hand inequality for all  $\omega$ , leaving us with

$$\Delta t \leq \frac{2}{\sqrt{z_{\max}}}. \quad (24)$$

### 3.2.2 First Order in Time Schemes

Continuing to use a central difference for the second derivative but only a one sided difference (forward or backward) for the first derivative in time yields two alternative first-order time schemes.

**Forward Difference** Using forward differences in time yields the scheme

$$\begin{aligned} & \frac{u(x, t + \Delta t) - 2u(x, t) + u(x, t - \Delta t)}{\Delta t^2} \\ & + a \frac{u(x, t + \Delta t) - u(x, t)}{\Delta t} = -\nabla E(x, t), \end{aligned}$$

which leads to the update formula

$$u^{n+1}(x) = \frac{(2 + a\Delta t) u^n - u^{n-1} - \Delta t^2 \nabla E^n}{1 + a\Delta t}. \quad (25)$$

Von Neumann analysis applied to the linearized homogeneous part of (23) yields the quadratic equation

$$\underbrace{(1 + a\Delta t)}_A \xi^2 + \underbrace{\left(\Delta t^2 z(\omega) - 2 - a\Delta t\right)}_B \xi + \underbrace{1}_C = 0$$

for the update amplification factor  $\xi = \xi(\omega)$ . Since  $1 + a\Delta t > 1$  for all positive  $a$  and  $\Delta t$ , the first condition  $A \geq C$  of the root amplitude lemma (for positive  $A$ ) is always satisfied. We may therefore restrict our attention to the second condition  $A + C \geq |B|$ , assuming real  $z(\omega)$ , to determine whether  $|\xi(\omega)| \leq 1$ . This gives the condition  $(1 + a\Delta t) + 1 \geq |\Delta t^2 z(\omega) - (2 + a\Delta t)|$  which is equivalent to  $0 \leq \Delta t^2 z(\omega) \leq 2(2 + a\Delta t)$ . In the case where  $z(\omega) \geq 0$ , we automatically satisfy the left-hand inequality for all  $\omega$ , which leaves us with  $z(\omega) \Delta t^2 - 2a\Delta t - 4 \leq 0$ . Plugging in the extreme case  $z_{\max}$  and restricting  $\Delta t$  to lie below the positive root in order to keep the quadratic expression on the left negative yield

$$\Delta t \leq \sqrt{\frac{4}{z_{\max}} + \left(\frac{a}{z_{\max}}\right)^2} + \frac{a}{z_{\max}}. \quad (26)$$

Notice that the CFL condition (24) for the second-order (central difference) scheme is sufficient but not necessary. If, however, we wish to obtain a condition independent of the damping  $a$ , then minimizing the upper bound with respect to  $a$  (by plugging in  $a = 0$ ) recovers this prior second-order CFL condition.

**Backward Difference** Using backward differences in time yields

$$\begin{aligned} & \frac{u(x, t + \Delta t) - 2u(x, t) + u(x, t - \Delta t)}{\Delta t^2} \\ & + a \frac{u(x, t) - u(x, t - \Delta t)}{\Delta t} = -\nabla E(x, t), \end{aligned}$$

which has the corresponding update

$$\begin{aligned} u^{n+1}(x) = & (2 - a\Delta t) u^n - (1 - a\Delta t) u^{n-1} \\ & - \Delta t^2 \nabla E^n. \end{aligned} \quad (27)$$

Similar analysis yields the quadratic equation

$$\xi^2(\omega) - \left(2 - a\Delta t - \Delta t^2 z(\omega)\right) \xi(\omega) + (1 - a\Delta t) = 0$$

for the amplification factor  $\xi(\omega)$ . The first condition  $A \geq C$  of the lemma (for positive  $A$ ) is always satisfied ( $1 \geq 1 - a\Delta t$ ) for all positive values of  $a$  and  $\Delta t$ . The second condition  $A + C \geq |B|$ , assuming real  $z(\omega) \geq 0$ , of the lemma, can

be expressed as  $\Delta t^2 z(\omega) + 2a\Delta t - 4 \leq 0$ . Plugging in the extreme case  $z_{\max}$  and restricting  $\Delta t$  to lie below the positive root in order to keep the quadratic expression on the left negative give the following CFL condition

$$\Delta t \leq \sqrt{\frac{4}{z_{\max}} + \left(\frac{a}{z_{\max}}\right)^2} - \frac{a}{z_{\max}}. \quad (28)$$

Notice that the CFL condition (24) for the central difference scheme is necessary (easily seen by applying the triangle inequality) but no longer sufficient as in the forward difference case. Furthermore, the constraint becomes increasingly restrictive as the damping coefficient  $a$  increases, making it impossible to formulate a sufficient damping-independent stability constraint. We will therefore give no further consideration to this scheme.

### 3.3 Recursive Increments and Properties of Explicit Schemes

For greater convenience in implementation, especially when upgrading existing gradient descent routines structured according to (20) with one array to store the evolving iterate  $u^n$  and another for its increment  $\Delta u^n$ , the explicit accelerated PDE discretizations can be expressed in terms of recursively defined increments. We give the explicit formulas in Eqs. (29)–(31), where  $\Delta u^{n-1}$  denotes the previously increment (kept in just one more added array).<sup>5</sup>

47mm

Summary of explicit schemes

$$\text{gradient descent: } \Delta u^n = -\Delta t \nabla E^n, \quad \Delta t \leq \frac{2}{z_{\max}} \quad (29)$$

$$\text{1-order accelerated: } \Delta u^n = \frac{1}{1+a\Delta t} \Delta u^{n-1} - \frac{\Delta t^2}{1+a\Delta t} \nabla E^n, \quad \Delta t \leq \sqrt{\frac{4}{z_{\max}} + \left(\frac{a}{z_{\max}}\right)^2} + \frac{a}{z_{\max}} \quad (30)$$

$$\text{2-order accelerated: } \Delta u^n = \frac{2-a\Delta t}{2+a\Delta t} \Delta u^{n-1} - \frac{2\Delta t^2}{2+a\Delta t} \nabla E^n, \quad \Delta t \leq \frac{2}{\sqrt{z_{\max}}} \quad (31)$$

Here we see more directly the traditional momentum style structure (i.e., heavy ball) in that the next increment  $\Delta u^n$  is expressed as a weighted combination of the gradient  $\nabla E^n$  and the previous increment  $\Delta u^{n-1}$ . Recursion (30) is equivalent to the first order in time, explicit update (25) using forward differences while the recursion (31) is equivalent to the second order in time, explicit update (23) using central differences, and as such they must adhere to the same CFL conditions (26) and (24) derived earlier for these corresponding schemes.

<sup>5</sup> For completeness, the first-order backward difference scheme can also be written recursively in the form  $\Delta u^n = (1 - a\Delta t) \Delta u^{n-1} - \Delta t^2 \nabla E^n$ .

#### 3.3.1 The First-Order Scheme as a Sub-Case of the Second-Order Scheme

For any choice of damping  $\alpha_1$  and time step  $\Delta t_1$  parameters used in the first-order scheme (denoted by subscript 1), we may obtain equivalent update iterations by substituting the following change of parameters into the second-order scheme (denoted by subscript 2)

$$\Delta t_2 = \frac{\Delta t_1}{\sqrt{1 + \frac{a_1 \Delta t_1}{2}}} \quad \text{and} \quad a_2 = \frac{a_1}{\sqrt{1 + \frac{a_1 \Delta t_1}{2}}}. \quad (32)$$

This is easily shown by algebraic simplification of the second-order update (and stability condition) after applying the change of parameters. The simplified result will yield the first-order scheme (and stability condition) in the original damping and time step parameters. In short, the first-order scheme is always equivalent to the second-order scheme with a reduced time step and damping via the contraction factor

$$\frac{1}{\sqrt{1 + \frac{a_1 \Delta t_1}{2}}} < 1.$$

A particular special case of this equivalency arises in considering the maximal stable time step for both schemes. For a fixed choice of damping  $a$ , the first-order scheme appears to allow a more generous upper bound than the second-order scheme. However, there is no effective difference when substituting (32) into the second-order scheme. Although the

upper bound on the time step is smaller, the contracted time step is also smaller, such that the maximum stable time step in the first-order scheme rescales exactly to the maximum stable time step in the second-order scheme. Thus, so long as the damping is also contracted according to (32), the first-order scheme implemented with its maximum stable time step is equivalent to the second-order scheme implemented with its maximum stable time step.

We may also consider the backward version of the change of parameters (32) in order to map the second-order scheme into the first-order scheme. In this case, using parameters  $\alpha_2$  and  $\Delta t_2$  in the second-order scheme is equivalent to applying the following change of parameters to the first-order scheme

$$\Delta t_1 = \frac{\Delta t_2}{\sqrt{1 - \frac{a_2 \Delta t_2}{2}}} \quad \text{and} \quad a_1 = \frac{a_2}{\sqrt{1 - \frac{a_2 \Delta t_2}{2}}}. \quad (33)$$

However, this backward mapping only applies when the second-order parameters satisfy  $a_2 \Delta t_2 < 2$ . When this condition is satisfied, we can show by direct substitution and algebraic simplification that the second-order scheme (and stability condition) is equivalent to the first-order scheme (and stability condition) with an amplified time step and damping coefficient via the amplification factor

$$\frac{1}{\sqrt{1 - \frac{a_2 \Delta t_2}{2}}} > 1.$$

Assuming the same condition is satisfied, the second-order scheme implemented with its maximum stable time step is equivalent to the first-order scheme implemented with its maximum stable time step after boosting the damping parameter according to (33).

### 3.3.2 Critical Damping in the Second-Order Scheme (Gradient Descent)

Unlike the forward mapping of the first order into the second-order discrete scheme, which is always possible for any choice of first-order discrete parameters  $a_1$  and  $\Delta t_1$ , the backward mapping is not possible for certain choices of the second-order discrete parameters, namely for  $a_2 \Delta t_2 \geq 2$ , where the backward amplification factor  $\frac{1}{\sqrt{1 - \frac{a_2 \Delta t_2}{2}}}$  is undefined. While  $\Delta t_2$  is upper bounded by the second-order scheme's stability constraint, there is no such upper bound imposed on  $a_2$  since the stability constraint is independent of  $a_2$ . As such for any stable, nonzero, second-order discrete time step  $\Delta t_2$ , we may always choose the second-order discrete damping coefficient  $a_2$  high enough to enter into this parameter regime where  $a_2 \Delta t_2 \geq 2$ . In this case the second-order scheme will exhibit behavior that is no longer reproducible by the first-order scheme.

It is interesting to consider what happens at the transition point when  $a_2 \Delta t_2 = 2$ . It is immediately seen, by plugging this into (31), that the second-order scheme becomes identical to the discrete gradient descent scheme (29) with an effective gradient descent time step of  $\Delta t = \frac{1}{2} \Delta t_2^2$  at this transition point (and if the second-order accelerated time step  $\Delta t_2$  was chosen to be the maximum stable step size of  $2/\sqrt{z_{\max}}$ , the effective gradient descent time step  $\Delta t$  will also be the maximum stable gradient descent step size of  $2/z_{\max}$ ). If we fix the second-order step size  $\Delta t_2$  and approach the transition point  $a_2 = 2/\Delta t_2$  from below, where an equivalent first-order damping coefficient  $a_1$  can be obtained via (33), then we see that the damping in the matching first-order scheme becomes infinite as the damping in the second-order

scheme approaches this critical value. This constitutes a discrete analog of the continuum property that the continuous gradient descent PDE (19) arises as the infinite frictional limit of the continuous accelerated descent PDE (18).

If we want a damping value  $a_2$  in the second-order scheme that will always keep us below this transition point for all choices of stable time step, then we must satisfy the inequality  $a_2 \Delta t_2 < 2$  for the maximum stable step size of  $2/\sqrt{z_{\max}}$ . This leads to the following upper bound for the second-order damping coefficient.

$$a_2 < \sqrt{z_{\max}}.$$

Namely, the square of damping factor should be strictly less than the gradient amplifier.

### 3.3.3 Over-Damping in the Second-Order Scheme (Gradient Descent with Resistance)

Noting that gradient descent arises in both schemes (although only in the limiting sense for the first-order scheme) at the transition point when  $a_2 \Delta t_2 = 2$  and that both schemes offer equivalent discretizations of accelerated descent according to the rescalings (32) and (33) below this transition point, it is now interesting to consider what happens above this transition point in the second-order scheme. If we choose  $a_2 > \sqrt{z_{\max}}$ , then there will be stable time step choices for  $\Delta t_2$  that will bring us beyond this transition.

In the case  $a_2 \Delta t_2 > 2$ , the second-order update in its recursive form (31) becomes a weighted combination of a step in the negative gradient direction as well as a *backward* step in the previous update direction. As such, the combined step can be interpreted as *partially undoing* the previous step, thereby slowing down the descent process. If we take the limiting case as the second-order damping coefficient  $a_2$  becomes infinite (keeping the same fixed time step  $\Delta t_2$ ), the stability of the scheme will not be affected, but the new update will fully subtract the previous update, thereby returning to the previous state before applying the new gradient step. Furthermore, after subtracting the previous update the amount of movement along the new gradient step will be zero for infinite  $a_2$ . This can be seen by noting that the weight on the previous update in (31) approaches  $-1$  from above and that the weight on the gradient approaches  $0$  from below as  $a_2 \rightarrow \infty$ . Therefore, in the limit, even if we initialize the recursion (31) with a nonzero starting update  $\Delta u^0$  (the discrete analog of an initial velocity), the effect will still be to remain motionless at the initial condition  $u^0$ .

This leads to the interpretation of the over-damped case as a *resisted* version of gradient descent for any finite  $a_2 > 2/\Delta t_2$ , since we start with a gradient step in the first update, then partially undo it before taking a new gradient step in the second update, which is then partially undone before taking

a new gradient step in the third update, and so on. Since the fraction of each gradient step which gets subtracted in the subsequent step remains fixed, rather than accumulating, we do not refer to this as deceleration but rather as *resistance*, which impedes the normal progress of gradient descent by a constant factor. As  $a_2$  increases, resistance increases, further slowing the progress of gradient descent, while completely halting it in the limit as  $a_2 \rightarrow \infty$ .

### 3.4 Semi-Implicit Schemes

We may use semi-implicit Euler style discretizations of (18) to obtain systems which more closely resemble the classic two-part Nesterov recursion. We may do this with any of the fully explicit schemes (23), (25), or (27) by replacing the explicit discretization  $\nabla E^n$  of the gradient with a “predicted estimate”  $\widehat{\nabla E}^{n+1}$  of its implicit discretization  $\nabla E^{n+1}$ . This estimate is obtained by applying the same discretization of  $\nabla E$  used in approximating  $\nabla E^n \approx \nabla E(u^n)$  to a partial update  $v^n$  for the “look ahead” approximation  $\widehat{\nabla E}^{n+1} \approx \nabla E(v^n)$ . The partial update  $v^n$  is obtained beforehand via the fully explicit update without the gradient term (i.e., by treating  $\nabla E^n$  as if it were zero). Using this strategy with the second order in time scheme (23) yields the two-step update

$$\begin{aligned} v^n &= u^n + \frac{2 - a\Delta t}{2 + a\Delta t} \Delta u^{n-1} \\ u^{n+1} &= v^n - \frac{2\Delta t^2}{2 + a\Delta t} \underbrace{\nabla E(v^n)}_{\approx \widehat{\nabla E}^{n+1}}. \end{aligned} \quad (34)$$

Notice the first and second steps, in isolation, both have a fully explicit structure. Von Neumann analysis can be employed to analyze the stability of this scheme according to the following update relationships between the DFT sequences  $U^n$ ,  $V^n$ , and  $U^{n+1}$  (transforms of  $u^n$ ,  $v^n$ , and  $u^{n+1}$ , respectively) where  $z(\omega)$  represents the gradient amplifier (21) associated with the linearization of  $\nabla E^n$  (and therefore also with the linearization of  $\widehat{\nabla E}^{n+1}$ ). We obtain

$$\begin{aligned} V^n &= \frac{4}{2 + a\Delta t} U^n - \frac{2 - a\Delta t}{2 + a\Delta t} U^{n-1} \\ U^{n+1} &= \left(1 - \frac{2\Delta t^2}{2 + a\Delta t} z(\omega)\right) V^n. \end{aligned}$$

If we substitute the first expression into the second, followed by substitutions  $U^{n\pm m} = \xi^{\pm m} U^n$ , then we obtain the quadratic equation

$$\underbrace{(2 + a\Delta t)^2}_{A} \xi^2 - \underbrace{4(2 + a\Delta t - 2\Delta t^2 z(\omega))}_{-B} \xi$$

$$+ \underbrace{(2 - a\Delta t)(2 + a\Delta t - 2\Delta t^2 z(\omega))}_{C} = 0,$$

for the overall combined update amplification factor  $\xi(\omega)$ . We may use the root amplitude lemma to check the stability criterion  $|\xi(\omega)| \leq 1$ .

#### First Stability Condition: $A \geq C$

The first condition from the lemma (for positive  $A$ ) can be expressed in quadratic form as  $az\Delta t^2 - (a^2 + 2z)\Delta t - 2a \leq 0$  which will be satisfied between its positive and negative roots. Restricting our interest to only positive values of  $\Delta t$  therefore yields the constraint

$$\Delta t \leq \frac{a^2 + 2z + \sqrt{(a^2 + 2z)^2 + 8a^2 z}}{2az} \underbrace{g(a, z) > 0}_{g(a, z) > 0}$$

where

$$\frac{\partial g}{\partial a} = \underbrace{(a^2 - 2z)}_{\text{toggles}} \left( \frac{a^2 + 2z + \sqrt{(a^2 + 2z)^2 + 8a^2 z}}{2a^2 z \sqrt{(a^2 + 2z)^2 + 8a^2 z}} \right) \underbrace{\text{always positive}}_{\text{always positive}}$$

To satisfy this independently of  $a$ , we examine the partial derivative of the upper bound  $g(a, z)$  with respect to  $a$  see that it starts out negative for  $a^2 < 2z$  and then turns positive for  $a^2 > 2z$ . The minimum upper bound is therefore attained when  $z(\omega) = z_{\max}$  and  $a^2 = 2z_{\max}$  yielding

$$\Delta t \leq \frac{2 + \sqrt{2}}{\sqrt{z_{\max}}}.$$

While this upper bound is more generous than (24) for the fully explicit scheme, it only satisfies the first of the two stability conditions in the bounded root lemma. We now proceed to the second condition which will be more restrictive.

#### Second Stability Condition: $A + C \geq |B|$

The second condition from the lemma (for positive  $A$ ) can be expressed as

$$\begin{aligned} 2 + a\Delta t - \Delta t^2 z(\omega) + \frac{1}{2} a\Delta t^3 z(\omega) \\ \geq \left| 2 + a\Delta t - 2\Delta t^2 z(\omega) \right|. \end{aligned}$$

For small enough time steps,  $2 + a\Delta t - 2\Delta t^2 z$  is positive, the absolute value signs can be removed, and the inequality holds. For larger time steps,  $2 + a\Delta t - 2\Delta t^2 z$  becomes and the inequality can be rearranged into the following cubic form

$$az(\omega)\Delta t^3 - 6z(\omega)\Delta t^2 + 4a\Delta t + 8 \geq 0.$$

Minimizing on the left with the case  $a = 0$  and  $z(\omega) = z_{\max}$  yields a stricter and therefore sufficient, stand-alone stability condition

$$\Delta t \leq \frac{2}{\sqrt{3}z_{\max}}. \quad (35)$$

Note that this upper bound is smaller, by a factor of  $\sqrt{3}$ , than the maximum stable time step (24) for the corresponding fully explicit scheme (23) or for its recursive equivalent (31).

## 4 Regularized Inversion via Accelerated PDEs

Here we consider a very general class of variational regularized inversion problems in the accelerated PDE framework. In particular, we assume energy functions with the form

$$E(u) = \int_{\Omega} \underbrace{f(|\mathcal{K}u - g|)}_{\text{fidelity}} + \underbrace{r(\|\nabla u\|)}_{\text{regularity}} \, dx$$

with  $\dot{f}, \dot{r}, \ddot{r} > 0$ . The function  $f$  is a monotonically increasing penalty on the residual error between data measurements  $g$  and a forward in the form of linear operator  $\mathcal{K}$  applied to the reconstructed signal  $u$ , while  $r$  is a monotonically increasing penalty on the gradient of the reconstruction.

### 4.1 General Case (Nonlinear Wave Equation)

The continuum gradient of  $E$  has the form

$$\begin{aligned} \nabla E(u) = & \underbrace{\frac{\dot{f}(|\mathcal{K}u - g|)}{|\mathcal{K}u - g|} \mathcal{K}^*(\mathcal{K}u - g)}_{\lambda(u,x) > 0} \\ & - \underbrace{\frac{\dot{r}(\|\nabla u\|)}{\|\nabla u\|} (\nabla \cdot \nabla u - u_{\eta\eta})}_{c(\nabla u) > 0} \\ & - \underbrace{\ddot{r}(\|\nabla u\|) u_{\eta\eta}}_{d(\nabla u) > 0} \end{aligned}$$

where  $\mathcal{K}^*$  denotes the adjoint of the forward operator  $\mathcal{K}$  and where  $\eta \doteq \frac{\nabla u}{\|\nabla u\|}$  denotes the unit vector along the gradient direction of  $u$ . This gives rise to the following class of accelerated flows which take the form of a nonlinear wave equation

$$\begin{aligned} u_{tt} - c(\nabla u) (\nabla \cdot \nabla u - u_{\eta\eta}) - d(\nabla u) u_{\eta\eta} + au_t \\ = \lambda(u, x) \mathcal{K}^*(g - \mathcal{K}u). \end{aligned}$$

If, purely for the sake of understanding stability, we model the short time behavior of any of the presented discrete update schemes in the neighborhood of a particular spatial point  $x$ , by treating  $\lambda$ ,  $c$ , and  $d$  as locally constant and by representing the forward model linear operator  $\mathcal{K}$  as a real convolution kernel  $K$  with adjoint  $K^T$ , then  $\nabla E$  can be approximated near  $x$  by the following linear expression

$$\begin{aligned} \nabla E \approx & \lambda_{[x]} K_{[x]}^T * K_{[x]} * (u^n - g) \\ & - c_{[x]} (\nabla \cdot \nabla u^n - u_{\eta\eta}) + d_{[x]} u_{\eta\eta} \end{aligned} \quad (36)$$

where the subscript  $[x]$  denotes the local point of spatially constant approximation (rather than a function argument). Assuming a uniform Cartesian grid oriented such that its first basis vector  $\mathbf{e}_1 = (1, 0, 0, \dots)$  aligns with  $\nabla u$  at our local point  $x$  and that our spatial derivative discretizations become equivalent to central difference (second derivative) approximations with space step  $\Delta x$  in each direction, then we obtain the following local approximation of the gradient amplifier of (36)

$$\begin{aligned} z(x, \omega_1, \omega_2, \dots, \omega_N) \\ \approx & \lambda_{[x]} \text{DFT}(K_{[x]}^T) \text{DFT}(K_{[x]}) \\ & + \frac{2}{\Delta x^2} \left( d_{[x]} (1 - \cos \omega_1 \Delta x) \right. \\ & \left. + c_{[x]} \sum_{k=2}^N (1 - \cos \omega_k \Delta x) \right). \end{aligned} \quad (37)$$

Noting that the Fourier transform of the adjoint  $K^T$  of a real convolution kernel is always the complex conjugate of the Fourier transform of the kernel  $K$  itself, we see that the gradient amplifier is real and positive and we can write the following upper bound as a function frequency  $\omega$

$$\begin{aligned} \max_{\omega} z \leq & \lambda_{[x]} \max_{\omega} |\text{DFT}(K_{[x]})|^2 \\ & + 4 \frac{c_{[x]} (N-1) + d_{[x]}}{\Delta x^2} \end{aligned}$$

with equality in cases where the complex amplitude of  $\text{DFT}(K)$  is maximal at  $\omega = (\pi, \dots, \pi)$ . However, since this upper bound depends on the local point of approximation  $x$ , we need to maximize over  $x$  as well in order to exploit the CFL formulas presented earlier in terms of  $z_{\max}$ . Doing so yields the following upper bound for the local gradient amplifier

$$z_{\max} \leq K_{\max} \lambda_{\max} + 4 \frac{(N-1)c_{\max} + d_{\max}}{\Delta x^2} \quad (38)$$

where  $\lambda_{\max} \doteq \max_x \lambda$ ,  $c_{\max} \doteq \max_x c$ ,  $d_{\max} \doteq \max_x d$ , and  $K_{\max} \doteq \max_{x,\omega} (|\text{DFT}(K)|^2)$ .

If we now plug (38) into the time step restriction (24) for the fully explicit second- order accelerated scheme (23), we obtain the following sufficient condition for stability

$$\Delta t \leq \frac{2\Delta x}{\sqrt{K_{\max}\lambda_{\max}\Delta x^2 + 4(N-1)c_{\max} + 4d_{\max}}}. \quad (39)$$

The corresponding condition for gradient descent is obtained by squaring  $\Delta x$  in the numerator and removing the radical (squaring) the denominator. As such, we note three favorable step size trends for PDE acceleration compared to PDE gradient descent. Most notably, when the regularizing coefficients  $c_{\max}$  and  $d_{\max}$  dominate, stable time step sizes are now directly proportional to spatial step sizes rather than to their squares, making the upper bound linear rather than quadratic in  $\Delta x$ . We see similar gains as well when the kernel  $K$  exhibits large amplification at one or more frequencies. In such cases, stable step sizes are inversely proportional to the maximum kernel amplification rather than to its square.

## 4.2 Quadratic Regularization (Linear Wave Equation)

The easiest special case to consider would be that of quadratic fidelity and regularity penalties without any forward model (more precisely with  $\mathcal{K}$  as the identity operator)

$$E(u) = \int_{\Omega} \frac{\lambda}{2} (u - g)^2 + \frac{c}{2} \|\nabla u\|^2 \, dx.$$

In this case, the gradient is linear and the local approximation (36) becomes exact with  $\lambda(x) = \lambda$ ,  $c(x) = d(x) = c$

$$\nabla E = \lambda(u - g) - c \nabla \cdot \nabla u.$$

The accelerated descent PDE therefore takes the form of a damped inhomogeneous linear wave equation.

$$u_{tt} - c \nabla \cdot \nabla u + au_t = \lambda(g - u). \quad (40)$$

In this case, the gradient amplifier  $z(\omega)$  (21) is easy to compute. If central differences on a uniform  $N$ -dimensional Cartesian grid with space step  $\Delta x$  in each direction are used to approximate the spatial derivatives of the Laplacian  $\nabla \cdot \nabla$ , then

$$z(\omega) = \lambda + \frac{2c}{\Delta x^2} \sum_{k=1}^N (1 - \cos \omega_k \Delta x),$$

where  $\omega = (\omega_1, \dots, \omega_N)$ , which makes the local approximation (37) exact as well. Its upper bound

$$z_{\max} = \lambda + \frac{4Nc}{\Delta x^2} \quad (41)$$

is attained at  $\omega = (\pi, \dots, \pi)$ , thereby making the general condition (39) necessary as well as sufficient for stability. Plugging all this into (29)–(31) yields the following fully explicit updates (and CFL conditions), with multi-index  $\alpha \in \mathbb{Z}^N$  to indicate each grid location, and where the additive multi-index

$$e_k = (\delta_{1k}, \delta_{2k}, \dots, \delta_{Nk})$$

is used to denote displacements to adjacent grid neighbors ( $\delta_{jk}$  being the standard Kronecker delta).

## Explicit fidelity schemes for quadratic regularization

$$\text{gradient descent} \quad \begin{cases} \Delta u_\alpha^n = -\Delta t \left( \lambda(u_\alpha^n - g_\alpha) - c \sum_{k=1}^N \frac{u_{\alpha+e_k}^n - 2u_\alpha^n + u_{\alpha-e_k}^n}{\Delta x^2} \right) \\ \Delta t \leq \frac{2\Delta x^2}{4Nc + \lambda \Delta x^2} \end{cases} \quad (42)$$

$$\text{1-order accelerated} \quad \begin{cases} \Delta u_\alpha^n = \frac{1}{1+a\Delta t} \Delta u_\alpha^{n-1} - \frac{\Delta t^2}{1+a\Delta t} \left( \lambda(u_\alpha^n - g_\alpha) - c \sum_{k=1}^N \frac{u_{\alpha+e_k}^n - 2u_\alpha^n + u_{\alpha-e_k}^n}{\Delta x^2} \right) \\ \Delta t \leq \Delta x \sqrt{\frac{4}{4Nc + \lambda \Delta x^2} + \left( \frac{a\Delta x}{4Nc + \lambda \Delta x^2} \right)^2} + \frac{a\Delta x^2}{4Nc + \lambda \Delta x^2} \end{cases} \quad (43)$$

$$\text{2-order accelerated} \quad \begin{cases} \Delta u_\alpha^n = \frac{2-a\Delta t}{2+a\Delta t} \Delta u_\alpha^{n-1} - \frac{2\Delta t^2}{2+a\Delta t} \left( \lambda(u_\alpha^n - g_\alpha) - c \sum_{k=1}^N \frac{u_{\alpha+e_k}^n - 2u_\alpha^n + u_{\alpha-e_k}^n}{\Delta x^2} \right) \\ \Delta t \leq \frac{2\Delta x}{\sqrt{4Nc + \lambda \Delta x^2}} \end{cases} \quad (44)$$

$$\text{semi-implicit} \quad \begin{cases} v_\alpha^n = u_\alpha^n + \frac{2-a\Delta t}{2+a\Delta t} \Delta u_\alpha^{n-1} \\ u_\alpha^{n+1} = v_\alpha^n - \frac{2\Delta t^2}{2+a\Delta t} \left( \lambda(v_\alpha^n - g_\alpha) - c \sum_{k=1}^N \frac{v_{\alpha+e_k}^n - 2v_\alpha^n + v_{\alpha-e_k}^n}{\Delta x^2} \right) \\ \Delta t \leq \frac{2\Delta x}{\sqrt{3(4Nc + \lambda \Delta x^2)}} \quad (\text{sufficient but not necessary when } a > 0) \end{cases} \quad (45)$$

## 4.3 Implicit Handling of the Fidelity Term

The portion of the continuum gradient which arises from the fidelity term is  $\lambda(u - g)$ , which we have discretized explicitly in the above schemes as  $\lambda(u_\alpha^n - g_\alpha)$ . Since this term, unlike the Laplacian discretization, does not depend upon neighboring

grid locations, we could evaluate it implicitly at the updated value of  $u$  by plugging  $\lambda(u_\alpha^{n+1} - g_\alpha)$  into any of these schemes and still rearrange the resulting expressions to obtain explicit updates for  $u_\alpha^{n+1}$ . Algebraic manipulation of these resulting implicitly handled fidelity schemes would yield the following equivalent schemes, restructured to reveal their similarity to the schemes (42)–(45).

## Implicit fidelity schemes for quadratic regularization

$$\text{gradient descent: } \Delta u_\alpha^n = -\frac{\Delta t}{1 + \lambda \Delta t} \left( \lambda(u_\alpha^n - g_\alpha) - c \sum_{k=1}^N \frac{u_{\alpha+e_k}^n - 2u_\alpha^n + u_{\alpha-e_k}^n}{\Delta x^2} \right) \quad (46)$$

$$\text{1-order accelerated: } \Delta u_\alpha^n = \frac{\Delta u_\alpha^{n-1} - \Delta t^2 \left( \lambda(u_\alpha^n - g_\alpha) - c \sum_{k=1}^N \frac{u_{\alpha+e_k}^n - 2u_\alpha^n + u_{\alpha-e_k}^n}{\Delta x^2} \right)}{1 + (a + \lambda \Delta t) \Delta t} \quad (47)$$

$$\text{2-order accelerated: } u_\alpha^n = \frac{(2 - a \Delta t) \Delta u_\alpha^{n-1} - 2 \Delta t^2 \left( \lambda(u_\alpha^n - g_\alpha) - c \sum_{k=1}^N \frac{u_{\alpha+e_k}^n - 2u_\alpha^n + u_{\alpha-e_k}^n}{\Delta x^2} \right)}{2 + a \Delta t + 2 \lambda \Delta t^2} \quad (48)$$

$$\text{semi-implicit: } \begin{cases} v_\alpha^n = u_\alpha^n + \frac{2-a\Delta t}{2+a\Delta t} \Delta u_\alpha^{n-1} \\ u_\alpha^{n+1} = v_\alpha^n - \frac{2\Delta t^2}{2+a\Delta t+2\lambda\Delta t^2} \left( \lambda(v_\alpha^n - g_\alpha) - c \sum_{k=1}^N \frac{v_{\alpha+e_k}^n - 2v_\alpha^n + v_{\alpha-e_k}^n}{\Delta x^2} \right) \end{cases} \quad (49)$$

Written in this form, it is easy to show by comparison that these schemes become equivalent to their explicit fidelity counterparts by a change of time step, damping parameter, or both. In the case of gradient descent, the implicit fidelity scheme (46) is identical to explicit fidelity scheme (42) with a smaller time step, using  $\Delta t \rightarrow \frac{\Delta t}{1+\lambda\Delta t}$ . The first-order implicit fidelity accelerated scheme (47) is equivalent to its explicit fidelity counterpart (43) with a larger damping coefficient, using  $a \rightarrow a + \lambda\Delta t$ . The second-order implicit fidelity accelerated scheme (48) is equivalent to the explicit fidelity scheme (44) with both a smaller time step and an adjusted damping coefficient (may be either larger or smaller depending on  $\lambda$ ), using  $\Delta t \rightarrow \frac{\Delta t}{\sqrt{1+\frac{\lambda}{2}\Delta t^2}}$  and  $a \rightarrow \frac{a+\lambda\Delta t}{\sqrt{1+\frac{\lambda}{2}\Delta t^2}}$ .

Finally, the implicit fidelity adaptation (49) of the semi-implicit scheme (45), obtained by replacing  $\lambda(u_\alpha^{n+1} - g)$  with  $\lambda(v_\alpha^n - g)$ , is equivalent to the original semi-implicit scheme (45) with both a smaller time step and a larger damping coefficient, using  $\Delta t \rightarrow \frac{\Delta t}{\sqrt{1+\frac{2\lambda\Delta t^2}{2+a\Delta t}}}$  and  $a \rightarrow a\sqrt{1+\frac{2\lambda\Delta t^2}{2+a\Delta t}}$ .

The CFL conditions for these implicit fidelity schemes can therefore be obtained by applying these substitutions backward to the matching explicit (or semi-implicit) CFL conditions. While this often yields a larger maximum stable time step, the apparent gain is deceptive since there will be no numerical difference to the corresponding explicit (or semi-implicit) update with a smaller time step. As such, there is neither a computational nor a numerical advantage to handling the fidelity term implicitly. While we have illustrated this here for the special case of quadratic regularization, the parameter remappings showing equivalency between the explicit and partially implicit schemes depend only upon the damping and fidelity parameters. It is easy to see that the exact same analysis applies even in the nonlinear case of non-quadratic regularization, making this equivalency (and therefore the lack of benefit in implicitly handling the fidelity) more general.

Further generalization of this analysis is also possible in the accelerated cases for non-quadratic fidelity penalization as well as for nontrivial forward models  $\mathcal{K}$ . However, in such cases, equivalency would require substitution of a constant damping parameter  $\alpha$  in the partially implicit scheme with a spatially varying damping in the equivalent explicit scheme. For example, in the case of a quadratic fidelity penalty paired with a convolution kernel  $K$  in the first-order accelerated implicit fidelity scheme (47), a constant damping parameter  $a$  would have to be replaced by the spatially varying  $a\mathcal{I} + \lambda\Delta t K^T K$  in order to use the explicit fidelity scheme (43) to obtain equivalent updates. This would require inversion of the matrix  $(1 + a\Delta t)\mathcal{I} + \lambda\Delta t^2 K^T K$ , as division by a scalar would no longer occur in the explicit update (43). However, since this inverse does not depend on  $u$ , its inverse could be computed/approximated just once and then reused in every

update step (in cases where the damping does not change with time).

#### 4.4 Beltrami Regularization

Another special case to consider is Beltrami regularization. We'll consider the case of a quadratic penalty and an attenuating, mean-preserving convolution kernel  $K$

$$E(u) = \int_{\Omega} \frac{\lambda}{2} (K * u - g)^2 + \underbrace{\frac{1}{\beta} \sqrt{1 + \|\beta \nabla u\|^2}}_{\sqrt{\epsilon^2 + \|\nabla u\|^2}, \epsilon = \frac{1}{\beta}} dx. \quad (50)$$

The gradient  $\nabla E$  is given by

$$\nabla E = \lambda K^T * K * (u - g) - \nabla \cdot \left( \underbrace{\frac{\beta \nabla u}{\sqrt{1 + \|\beta \nabla u\|^2}}}_{\frac{\nabla u}{\sqrt{\epsilon^2 + \|\nabla u\|^2}}, \epsilon = \frac{1}{\beta}} \right).$$

In this case, the nonlinear variational gradient decomposes as in (36) to the form

$$\begin{aligned} \nabla E = \lambda K^T * K * (u - g) - & \underbrace{\frac{\beta}{\sqrt{1 + \|\beta \nabla u\|^2}} (\nabla \cdot \nabla u - u_{\eta\eta})}_c \\ & - \underbrace{\frac{\beta}{(\sqrt{1 + \|\beta \nabla u\|^2})^3} u_{\eta\eta}}_d \end{aligned}$$

The accelerated PDE (technically an integral partial differential equation with the convolution) takes the quasilinear form

$$u_{tt} - \nabla \cdot \left( \frac{\beta \nabla u}{\sqrt{1 + \|\beta \nabla u\|^2}} \right) + au_t = \lambda K^T * K * (g - u). \quad (51)$$

Note that both coefficients  $c$  and  $d$  are bounded by  $\beta$  (an upper bound which is actually reached in both cases at any point and time where  $\nabla u(x, t) = 0$ ) and that  $\max |\text{DFT}(K)| = 1$  by our assumption that  $K$  attenuates while preserving the mean. Plugging this into (38) yields

$$z_{\max} \leq \lambda + \frac{4N\beta}{\Delta x^2} \quad (52)$$

if we assume a consistent discretization of

$$\nabla \cdot \left( \frac{\beta \nabla u}{\sqrt{1 + \|\beta \nabla u\|^2}} \right),$$

which converges, as  $\nabla u \rightarrow 0$ , to the central difference approximation of the  $\beta$ -scaled Laplacian  $\beta \nabla \cdot \nabla u \approx \beta \sum_{k=1}^N \frac{u_{\alpha+e_k}^n - 2u_{\alpha}^n + u_{\alpha-e_k}^n}{\Delta x^2}$  with spatial step size  $\Delta x$  in each direction (see Sect. 4.2 for the multi-index subscript notation  $\alpha$  and  $e_k$ ). If we let  $D_{\beta, \Delta x}^2 u^n$  denote the discretization of  $\nabla \cdot \left( \frac{\beta \nabla u}{\sqrt{1 + \|\beta \nabla u\|^2}} \right)$ , then we obtain the schemes in Eqs. (53)–(56).

| 92mm                | Summary of schemes for Beltrami regularization   |
|---------------------|--|
| gradient descent    | $\begin{cases} \Delta u^n = -\Delta t \left( \lambda K^T * K * (u^n - g) - D_{\beta, \Delta x}^2 u^n \right) \\ \Delta t \leq \Delta x^2 \left( \frac{2}{4N\beta + \lambda \Delta x^2} \right) \end{cases} \quad (53)$   |
| 1-order accelerated | $\begin{cases} \Delta u^n = \frac{1}{1+a\Delta t} \Delta u^{n-1} - \frac{\Delta t^2}{1+a\Delta t} \left( \lambda K^T * K * (u^n - g) - D_{\beta, \Delta x}^2 u^n \right) \\ \Delta t \leq \Delta x \left( \sqrt{\frac{4}{4N\beta + \lambda \Delta x^2}} + \left( \frac{a\Delta x}{4N\beta + \lambda \Delta x^2} \right)^2 + \frac{a\Delta x}{4N\beta + \lambda \Delta x^2} \right) \end{cases} \quad (54)$ |
| 2-order accelerated | $\begin{cases} \Delta u^n = \frac{2-a\Delta t}{2+a\Delta t} \Delta u^{n-1} - \frac{2\Delta t^2}{2+a\Delta t} \left( \lambda K^T * K * (u^n - g) - D_{\beta, \Delta x}^2 u^n \right) \\ \Delta t \leq \Delta x \left( \frac{2}{\sqrt{4N\beta + \lambda \Delta x^2}} \right) \end{cases} \quad (55)$   |
| semi-implicit       | $\begin{cases} v^n = u^n + \frac{2-a\Delta t}{2+a\Delta t} \Delta u^{n-1} \\ u^{n+1} = v^n - \frac{2\Delta t^2}{2+a\Delta t} \left( \lambda K^T * K * (v^n - g) - D_{\beta, \Delta x}^2 v^n \right) \\ \Delta t \leq \Delta x \left( \frac{2}{\sqrt{3(4N\beta + \lambda \Delta x^2)}} \right) \end{cases} \quad (56)$  |

## 4.5 Total Variation Regularization

If we consider the limit as  $\beta \rightarrow \infty$ , the Beltrami regularization penalty converges to the total variation penalty

$$E(u) = \int_{\Omega} \frac{\lambda}{2} (K * u - g)^2 + \|\nabla u\| \, dx \quad (57)$$

with a nonlinear variational gradient (36) that decomposes as

$$\begin{aligned} \nabla E &= \lambda K^T * K * (u - g) - \nabla \cdot \left( \frac{\nabla u}{\|\nabla u\|} \right) \\ &= \lambda K^T * K * (u - g) - \underbrace{\frac{1}{\|\nabla u\|}}_c (\nabla \cdot \nabla u - u_{\eta\eta}). \end{aligned}$$

The accelerated PDE now takes the form of the nonlinear wave equation

$$u_{tt} - \nabla \cdot \left( \frac{\nabla u}{\|\nabla u\|} \right) + au_t = \lambda K^T * K * (g - u). \quad (58)$$

In this case, the coefficient  $d$  vanishes, but the coefficient  $c$  no longer has a finite upper bound. Plugging this into (38) yields an infinite upper bound for the maximum gradient amplifier if at any point and time  $\nabla u(x, t) = 0$ . Otherwise, by our earlier assumption on  $K$  (see Sect. 4.4) we obtain

$$\lambda \leq z_{\max} \leq \lambda + \frac{4(N-1)}{\Delta x^2 \min \|\nabla u\|}. \quad (59)$$

For the explicit second-order accelerated scheme, this ensures the sufficient condition  $\Delta t \leq \frac{2}{\sqrt{\lambda + \frac{4(N-1)}{\Delta x^2 \min \|\nabla u\|}}}$  for a stable step. If we fix  $\Delta t$ , we may rearrange this inequality to obtain an equivalent sufficient condition

$$\min \|\nabla u\| \geq \frac{N-1}{\Delta x^2} \frac{4\Delta t^2}{4 - \lambda \Delta t^2}$$

which takes the form of a lower bound on the spatial gradient.

Here an interesting nonlinear dynamic occurs to keep the implementation stable by preventing initiated instabilities from growing unbounded. If the spatial gradient falls below this lower bound and instabilities begins to propagate at one or more frequencies, they will eventually cause the spatial gradient to rise above the guaranteed stable lower bound at which point the instabilities will cease growing. In the absence of a kernel  $K$ , the fastest growing instability will occur at the highest digital frequency in each grid

direction  $\omega = (\pi, \dots, \pi)$  which corresponds to oscillations between immediately adjacent grid points; this in turn will most rapidly increase the discrete difference approximations of  $\|\nabla u\|$ . In the presence of a strongly smoothing kernel, the fastest growing instability may occur at lower digital frequencies, thereby causing a low-grade ringing effect, with several grid points per period, until the amplitude of the oscillation is large enough to drive adjacent pixel differences back over the lower bound for  $\|\nabla u\|$ .

A similar phenomenon occurs with both the first-order and semi-implicit schemes (and even with gradient descent), making all these schemes stable independently of the regularizer coefficient  $c$ . As such, purely for stability considerations alone, the necessary step size constraint will be connected to the lower bound  $\lambda$  of the gradient amplifier  $z_{\max}$  rather than its upper bound in (59). This yields the following necessary conditions for stability. However, the schemes may only converge under these constraints in an oscillatory sense with a fluctuating level of “background noise” whose amplitude will depend upon the value of  $\Delta t$ .

$$\text{gradient descent: } \Delta t \leq \frac{2}{\lambda} \quad (60)$$

$$\text{1-order accelerated: } \Delta t \leq \sqrt{\frac{4}{\lambda} + \left(\frac{a}{\lambda}\right)^2} + \frac{a}{\lambda} \quad (61)$$

$$\text{2-order accelerated: } \Delta t \leq \frac{2}{\sqrt{\lambda}} \quad (62)$$

$$\text{semi-implicit: } \Delta t \leq \frac{2}{\sqrt{3\lambda}} \quad (63)$$

We may exploit the behavior of this nonlinear stabilizing effect to obtain a more useful time step constraint by plugging in a minimal acceptable value of  $\|\nabla u\|$  for the final reconstruction into the stability condition for  $\Delta t$ . A natural way to approach this is by exploiting a quantization interval  $Q$  for the digital representation of  $u$  together with the following discrete approximation bounds for  $\|\nabla u\|$

$$\begin{aligned} \min \|\nabla u\| &= \min_{\alpha} \sqrt{\sum_{k=1}^N \left( \frac{u_{\alpha+e_k} - u_{\alpha}}{\Delta x} \right)^2} \\ &\geq \sqrt{N \min_{\alpha,k} \left( \frac{u_{\alpha+e_k} - u_{\alpha}}{\Delta x} \right)^2} \\ &= \frac{\sqrt{N}}{\Delta x} \min_{\alpha,k} |u_{\alpha+e_k} - u_{\alpha}|. \end{aligned}$$

If we now determine that instability-related distortions confined to a single quantization interval  $Q$  between neighboring pixels are acceptable, we substitute

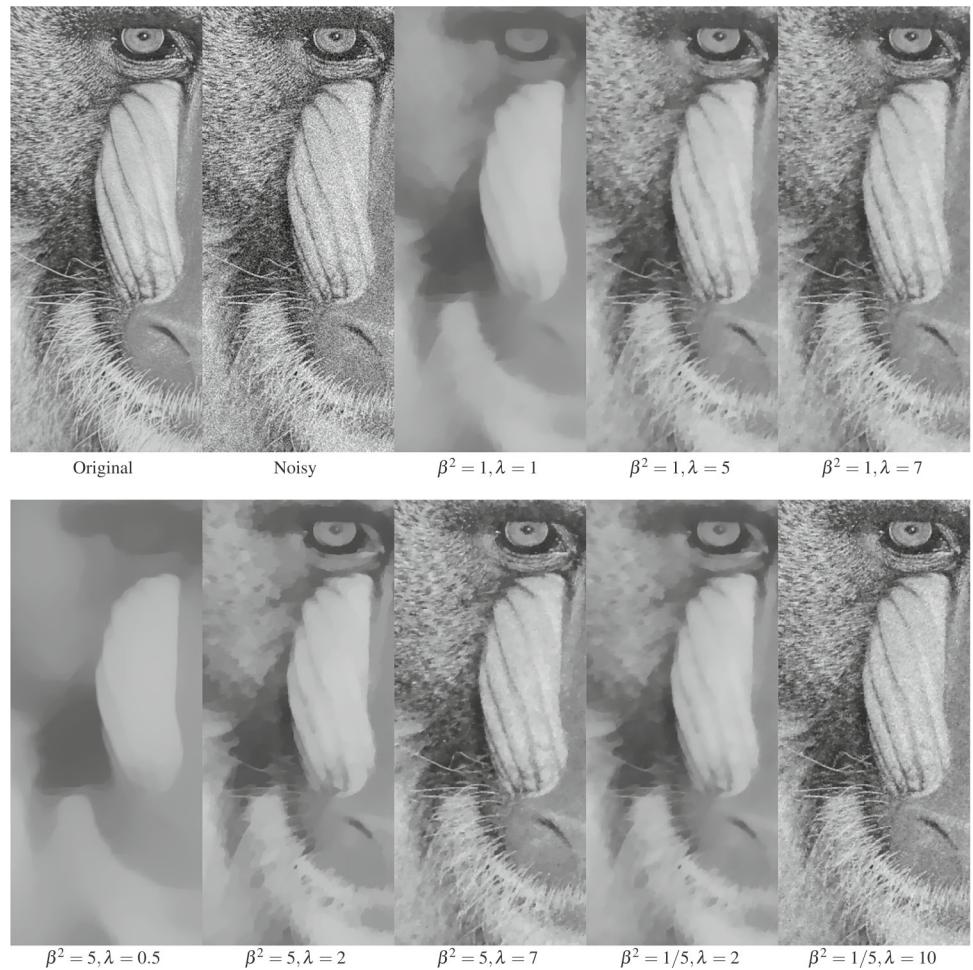
$$\min \|\nabla u\| \rightarrow \frac{\sqrt{N}}{\Delta x} Q$$

into the upper bound for (59) to obtain

$$z_{\max} \leq \lambda + \frac{4(N-1)}{Q \Delta x \sqrt{N}} < \lambda + \frac{4\sqrt{N}}{Q \Delta x} \quad (64)$$

within the desired stable regime for  $\|\nabla u\|$ . This in turn gives rise to the schemes in Eqs. (65)–(68), where  $D_{\Delta x}^2 u^n$  denotes the discretization of  $\nabla \cdot \left( \frac{\nabla u}{\|\nabla u\|} \right)$ .

**Fig. 2** Results of Beltrami regularization applied to a noisy baboon image with varying values of  $\lambda$  and  $\beta$ . The units of  $\lambda$  are thousands



**Table 1** PDE accelerated Beltrami regularization runtimes on the  $512 \times 512$  baboon image

|                 | $\lambda = 1000$ |            | $\lambda = 5000$ |            | $\lambda = 7000$ |            |
|-----------------|------------------|------------|------------------|------------|------------------|------------|
|                 | Time (s)         | Iterations | Time (s)         | Iterations | Time (s)         | Iterations |
| $\beta^2 = 1/5$ | 0.55             | 124        | 0.27             | 60         | 0.23             | 50         |
| $\beta^2 = 1$   | 0.81             | 183        | 0.38             | 85         | 0.32             | 71         |
| $\beta^2 = 5$   | 1.20             | 273        | 0.54             | 122        | 0.45             | 101        |

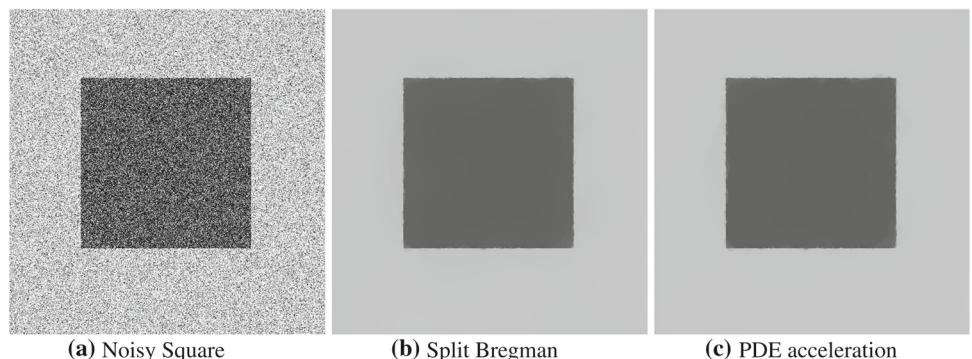
**Fig. 3** An example of inpainting using the PDE accelerated Beltrami regularization framework on the cameraman image

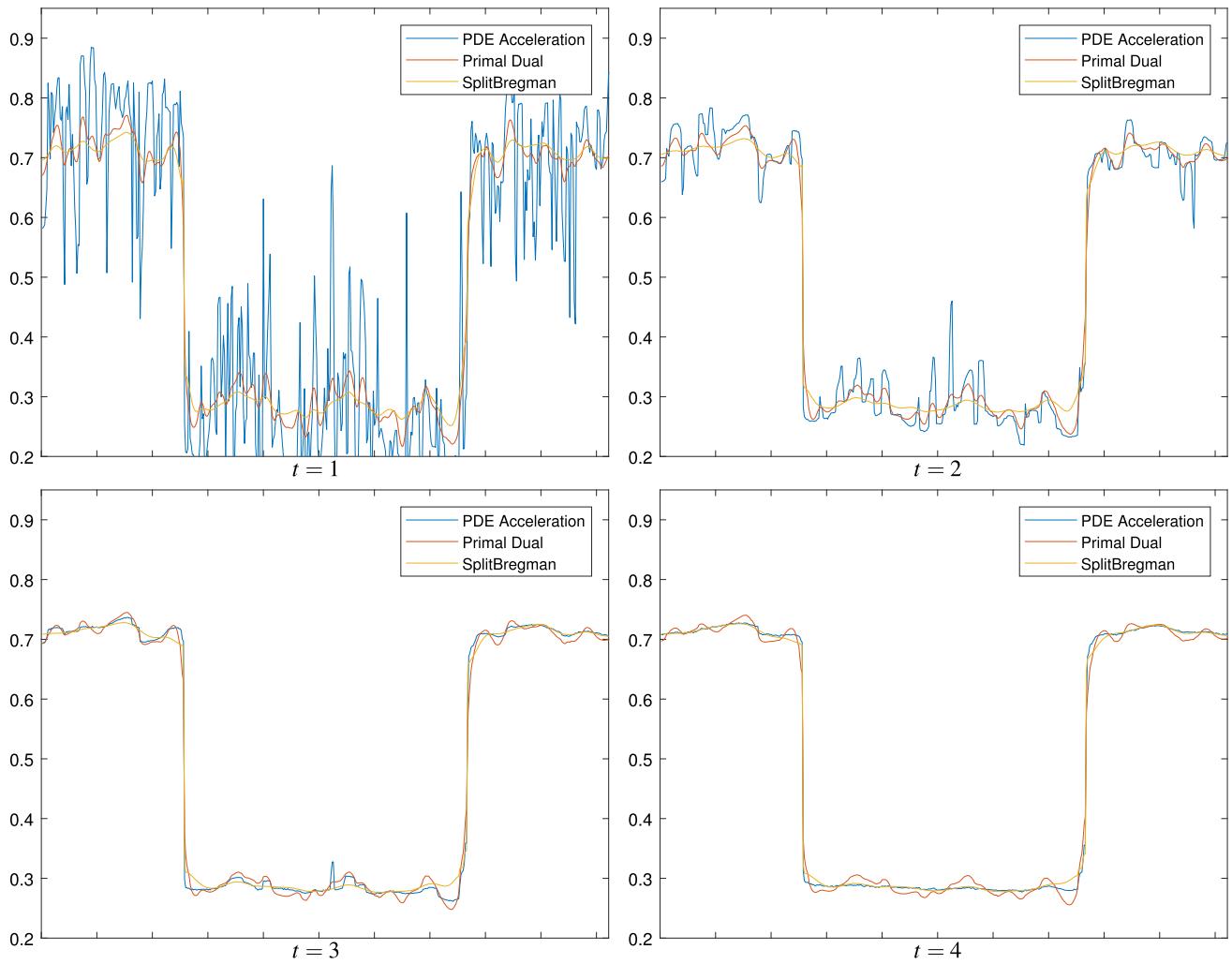


**Fig. 4** Deblurring of an image using the explicit accelerated PDE scheme compared with the results of two other state-of-the-art methods (final signal-to-noise ratios shown for each restoration)

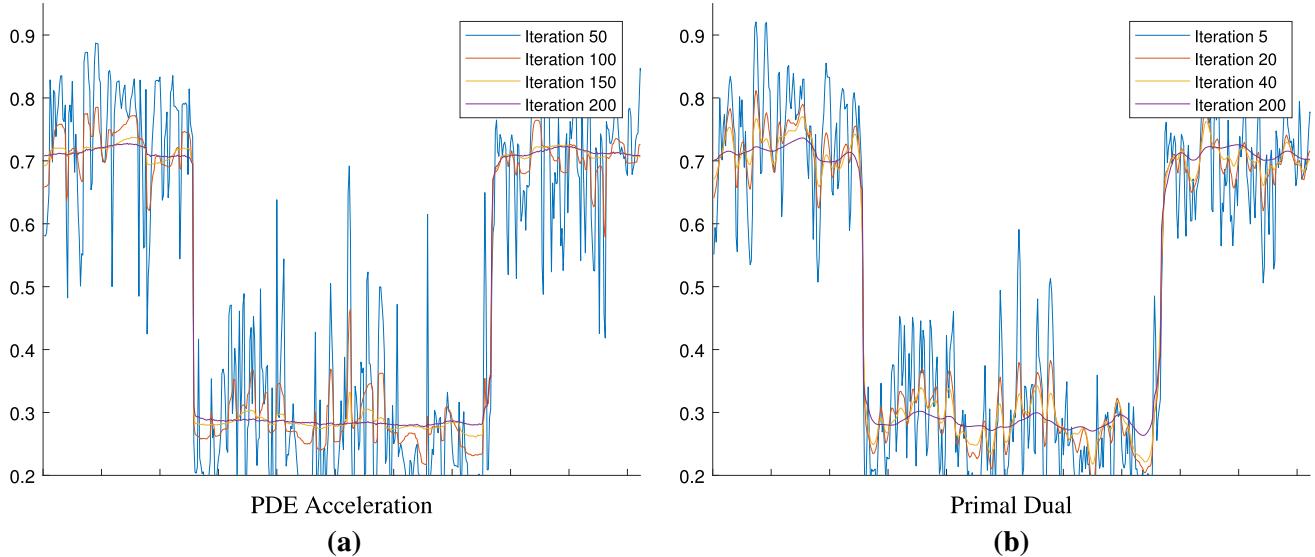


**Fig. 5** Denoising of a synthetic image with total variation restoration with  $\lambda = 1000$  via **b** split Bregman and **c** PDE acceleration. In PDE acceleration, we used  $\Delta t = \Delta x/2$  and  $a = 2\sqrt{\lambda}$





**Fig. 6** Comparison of PDE acceleration, primal–dual, and split Bregman algorithms for denoising a noisy square image. A one-dimensional slice of the image is displayed at the same computation time for each algorithm



**Fig. 7** Comparison of flows generated by **a** PDE acceleration and **b** primal–dual for solving the TV restoration problem on the noisy square image. Notice the edges are better preserved in PDE acceleration earlier in the flow

## Summary of schemes for Total Variation regularization

$$\text{gradient descent} \quad \begin{cases} \Delta u^n = -\Delta t (\lambda K^T * K * (u^n - g) - D_{\Delta x}^2 u^n) \\ \Delta t \leq Q \Delta x \left( \frac{2}{4\sqrt{N} + \lambda Q \Delta x} \right) \end{cases} \quad (65)$$

$$\text{1-order accelerated} \quad \begin{cases} \Delta u^n = \frac{1}{1+a\Delta t} \Delta u^{n-1} - \frac{\Delta t^2}{1+a\Delta t} (\lambda K^T * K * (u^n - g) - D_{\Delta x}^2 u^n) \\ \Delta t \leq \sqrt{Q \Delta x} \left( \sqrt{\frac{4}{4\sqrt{N} + \lambda Q \Delta x}} + \left( \frac{a\sqrt{Q \Delta x}}{4\sqrt{N} + \lambda Q \Delta x} \right)^2 + \frac{a\sqrt{Q \Delta x}}{4\sqrt{N} + \lambda Q \Delta x} \right) \end{cases} \quad (66)$$

$$\text{2-order accelerated} \quad \begin{cases} \Delta u^n = \frac{2-a\Delta t}{2+a\Delta t} \Delta u^{n-1} - \frac{2\Delta t^2}{2+a\Delta t} (\lambda K^T * K * (u^n - g) - D_{\Delta x}^2 u^n) \\ \Delta t \leq \sqrt{Q \Delta x} \left( \sqrt{\frac{4}{4\sqrt{N} + \lambda Q \Delta x}} \right) \end{cases} \quad (67)$$

$$\text{semi-implicit} \quad \begin{cases} v^n = u^n + \frac{2-a\Delta t}{2+a\Delta t} \Delta u^{n-1} \\ u^{n+1} = v^n - \frac{2\Delta t^2}{2+a\Delta t} (\lambda K^T * K * (v^n - g) - D_{\Delta x}^2 v^n) \\ \Delta t \leq \sqrt{Q \Delta x} \left( \sqrt{\frac{4}{3(4\sqrt{N} + \lambda Q \Delta x)}} \right) \end{cases} \quad (68)$$

## 5 Experimental Examples

### 5.1 Beltrami Denoising

Our first application is to the problem of Beltrami regularization for image denoising and image inpainting [15, 21, 32], which corresponds to minimizing (50) in the absence of a kernel  $K$  via the accelerated PDE (51). In this case,  $g$  is the original noisy image, and the minimizer  $u$  is the denoised/inpainted image. For denoising, we typically set the parameter  $\lambda$  to be a positive constant, and for inpainting we can set  $\lambda = 0$  in the region  $D \subset \Omega$  to be inpainted and set  $\lambda$  to be large or  $\infty$  in  $\Omega \setminus D$ . The Beltrami regularization term interpolates between the TV norm  $\int |\nabla u|$  and the  $H^1$  norm  $\int |\nabla u|^2$ —near edges where  $\nabla u$  is large, it behaves like the TV norm to preserve edges, and where  $\nabla u$  is small it behaves like the  $H^1$  norm in order to reduce staircasing. Recently, Zosso and Bustin [32] have proposed an efficient primal–dual projected gradient method for solving Beltrami regularized problems.

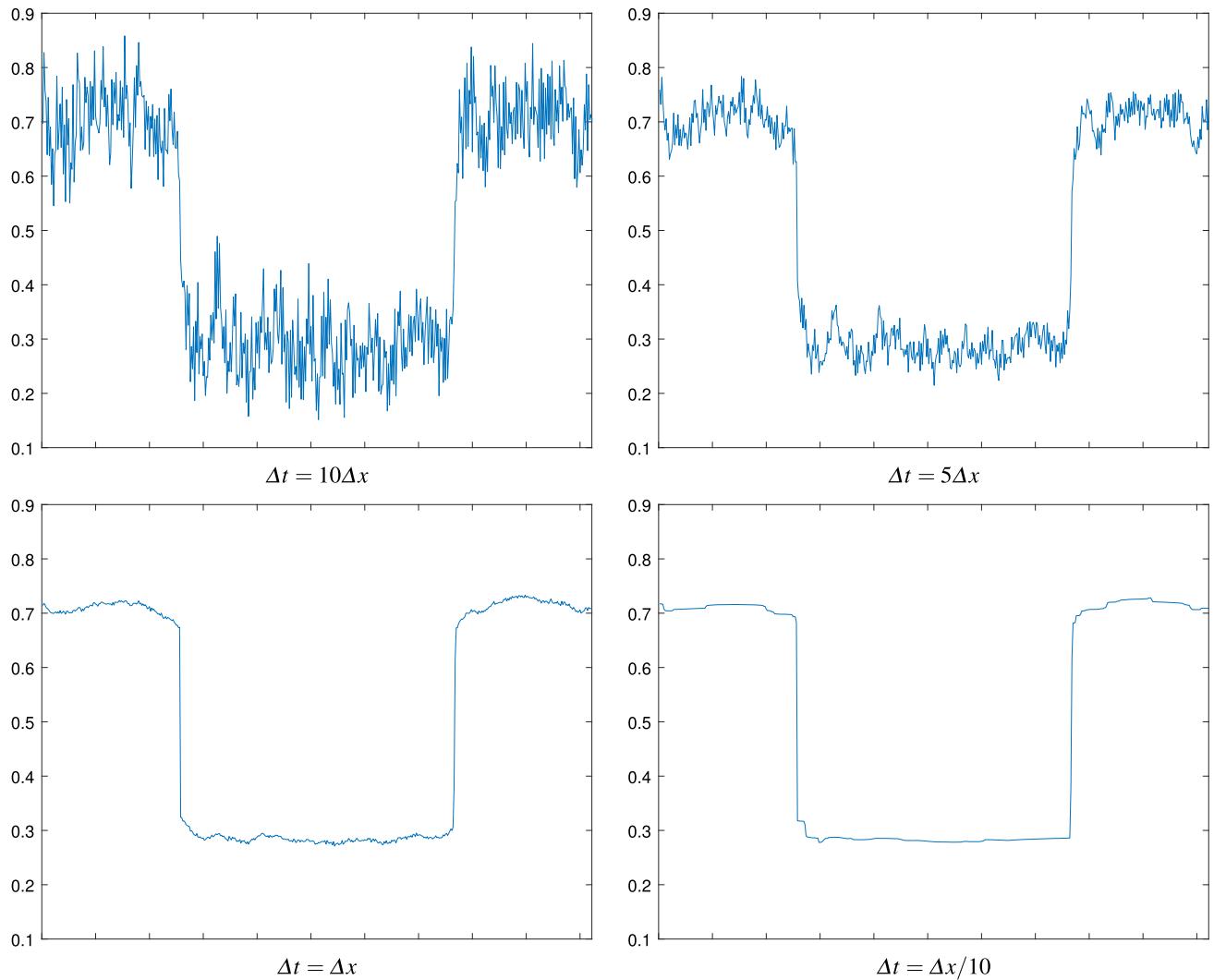
We use the first-order explicit scheme (54) with forward differences for  $\nabla u$  and backward differences for  $\text{div}$ . We set the damping coefficient to  $a = 2\sqrt{\beta\pi^2 + \lambda}$  via the linear analysis (16) and run the algorithm at its maximum stable step  $\Delta t$  (54) until the absolute difference between the current and previous iterates falls below  $10^{-4}$ . We note that the image is normalized so the pixel values fall in the interval  $[0, 1]$ .

Figure 2 shows the results of applying the PDE accelerated Beltrami regularization to a noisy baboon test image with varying values of  $\lambda$  and  $\beta$  with single-threaded C++

code on a 3.2-GHz Intel processor running Linux. The corresponding runtimes are given in Table 1 and are favorably competitive with the runtimes reported in [32], who proposed a primal–dual projected gradient algorithm for Beltrami regularization. Notice the algorithm does slow down somewhat when  $\lambda$  is small and the denoising is heavily regularized, but the difference is far less pronounced compared to other explicit methods such as gradient descent.

### 5.2 Beltrami Inpainting

We also give an example of PDE acceleration for Beltrami regularized inpainting in Fig. 3. We used  $\beta = 1$  and  $a = 5\pi$ , and the inpainting took 687 iterations (11.48 s) starting from an initial guess given by nearest neighbor interpolation. This is a good deal slower than the denoising examples. It is possible to give a partial explanation for this. Recall that the optimal damping parameter, and convergence rate, depends on the size of the first eigenvalue of the linearized operator on the given domain and the presence of a zeroth-order term  $\lambda u$ . In inpainting, there is no zeroth-order term and the domain is highly irregular. Further, the inpainting domain is typically disconnected, so the eigenvalues on each connected component would be required, and this would lead to different choices of damping coefficient in each region. We plan to investigate this issue, and others, in future work.



**Fig. 8** Comparison of steady-state solutions for denoising the 2D noisy square for different time steps in PDE acceleration. The scheme is stable in  $L^\infty$  for a variety of time steps though we observe  $\Delta t \leq \Delta x$  is required to ensure the solution is a reasonable denoising

### 5.3 Beltrami Deblurring

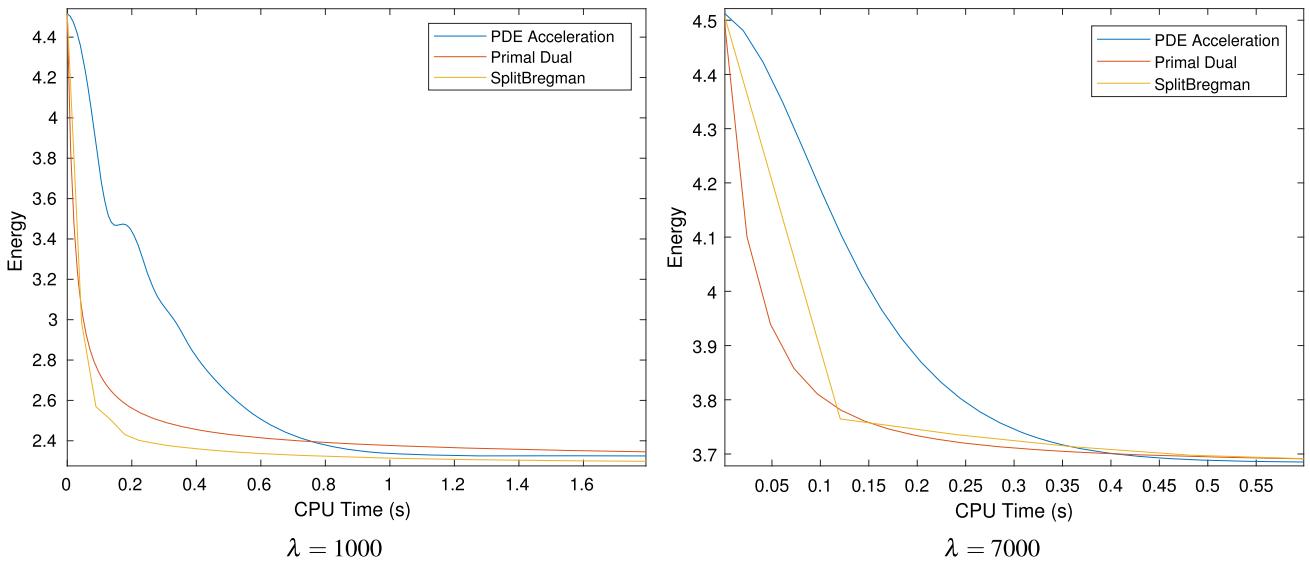
Finally, we give an example of PDE acceleration for Beltrami regularized deblurring. We used  $\lambda = 10^7$ ,  $\beta = 1$ , and  $a = 4$ , and the deblurring was run using the second-order explicit scheme (55) with its maximum stable time step starting with the original blurred image as the initial guess. After 2038 iterations, it achieved its tenth-of-a-decibel rounded steady-state restored PSNR of 32.3 dB. The original image was blurred with a Gaussian kernel of  $\sigma = 3$  to create an blurry initial image with a signal-to-noise ratio of 25.6185 dB. In Fig. 4, we compare the accelerated PDE results, both visually and quantitatively according to the restored signal-to-noise ratio, with those obtained using primal–dual and L1 ADMM algorithms for the same parameters  $\lambda = 10^7$  and  $\beta = 1$ . ADMM reached its tenth-of-a-decibel rounded steady-state restored PSNR of 31.8 dB after 2453 iterations, whereas primal–dual

reached its tenth-of-a-decibel rounded steady-state restored PSNR of 27.8 dB after 63 iterations (significantly fewer iterations than both other algorithms, but also significantly lower restored PSNR).

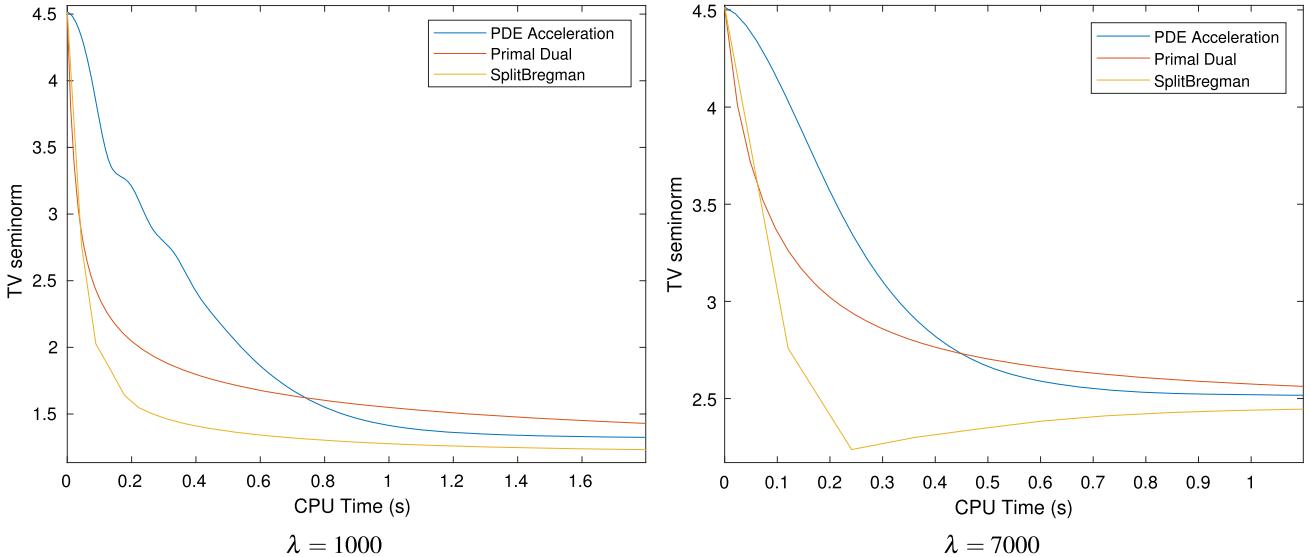
### 5.4 TV Denoising

We now consider the problem of total variation (TV) restoration, which has a long history in image processing [20]. The TV denoising problem corresponds to minimizing (57) in the absence of a kernel  $K$  via the accelerated PDE (58). In this case, state-of-the-art approaches include primal–dual methods [9] and the split Bregman method [12].

We again use the first-order explicit scheme (66), while discretizing the spatial gradient and divergence separately (using forward differences for the gradient and backward differences for the divergence), and homogeneous Neumann



**Fig. 9** Comparison of logarithm of total energy versus CPU time for denoising the full  $512 \times 512$  Lenna image with PDE acceleration, primal–dual, and split Bregman. We used the optimal damping  $a = 2\sqrt{\lambda}$  from the linear analysis



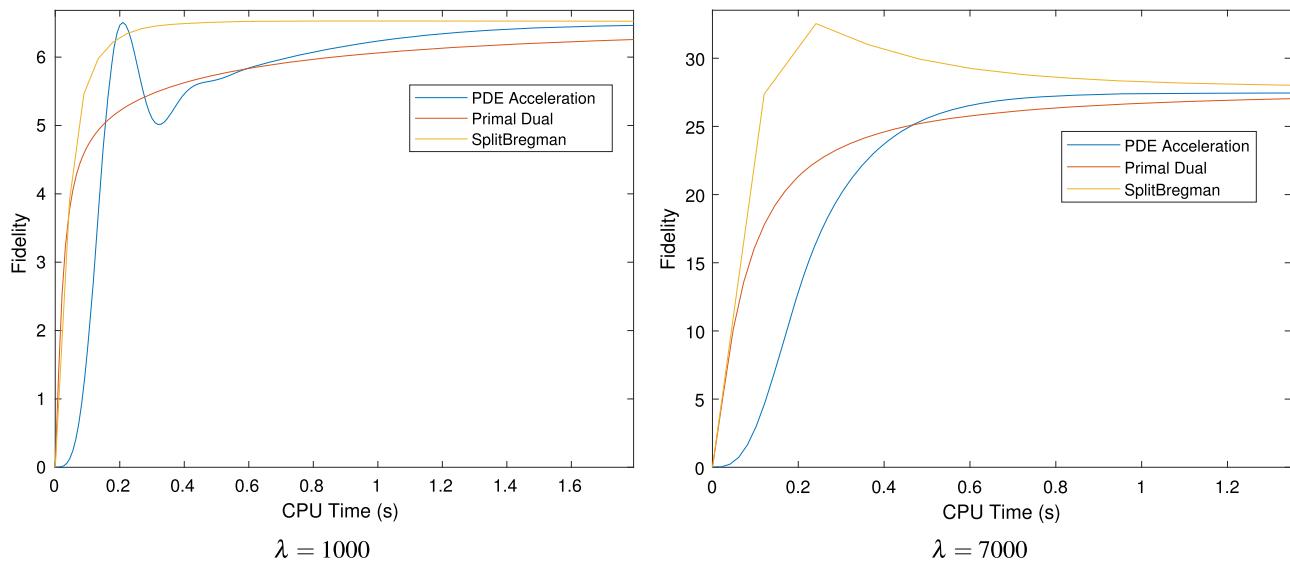
**Fig. 10** Comparison of logarithm of TV seminorm energy versus CPU time for denoising the full  $512 \times 512$  Lenna image with PDE acceleration, primal–dual, and split Bregman

boundary conditions. Numerically, we set  $\nabla u / |\nabla u| = 0$  whenever  $\nabla u = 0$ , so no regularization is required, though we rarely encounter numerical gradients that are identically zero. This choice of discretization makes the discrete divergence the exact numerical adjoint of the discrete gradient.

We first consider a noisy square image, with dark region  $u = 0.25$  and light region  $u = 0.75$  with additive Gaussian noise with standard deviation  $\sigma = 0.3$ . Figure 5 shows the noisy square and the total variation denoising with the split Bregman algorithm and PDE acceleration. We compare PDE acceleration, primal–dual, and split Bregman on slices of the image at similar computation times in Figs. 6 and 7.

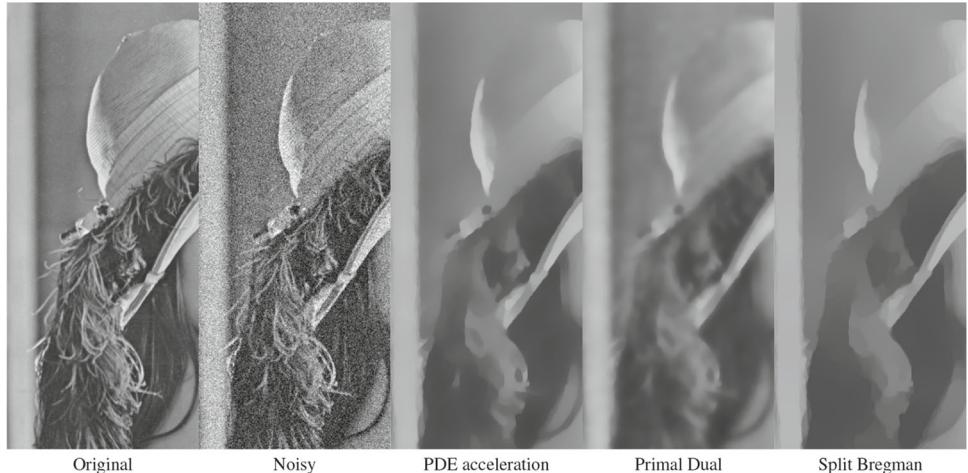
Notice the primal–dual algorithm blurs the edges slightly at first, and they are restored only late in the flow (at  $t = 4$  primal–dual has not yet converged). The PDE acceleration algorithm does a better job preserving edges (they are never blurred) compared to primal–dual and is slightly better than split Bregman at preserving edges by time  $t = 4$ .

In the example above, we took  $\Delta t = \Delta x/2$  for simplicity. Corroborating our analysis in Sect. 4.5, this explicit numerical scheme (66) behaves stably in  $L^\infty$  in our experiments, meaning the solutions remain bounded in  $L^\infty$  for all time, even for larger time steps which still satisfy the necessary conditions (60), (61), (62) or (63). For such larger time steps,

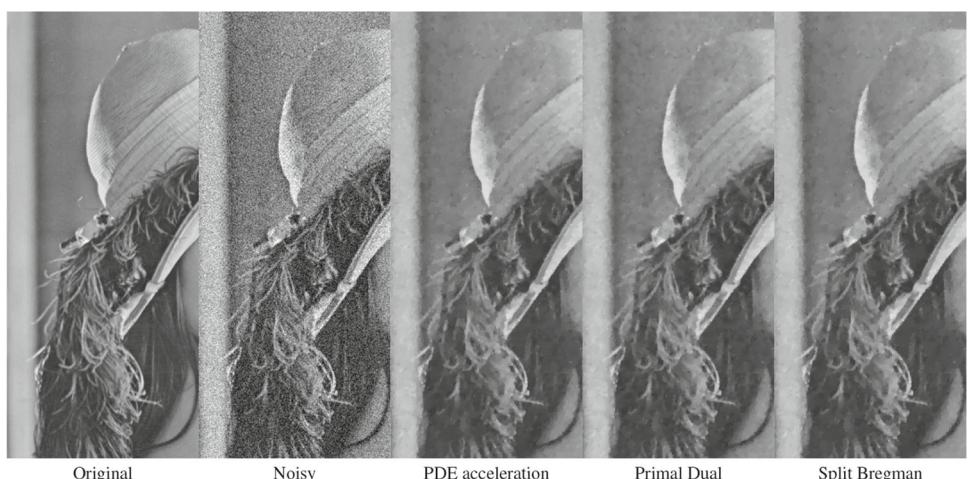


**Fig. 11** Comparison of fidelity energy versus CPU time for denoising the full  $512 \times 512$  Lenna image with PDE acceleration, primal–dual, and split Bregman

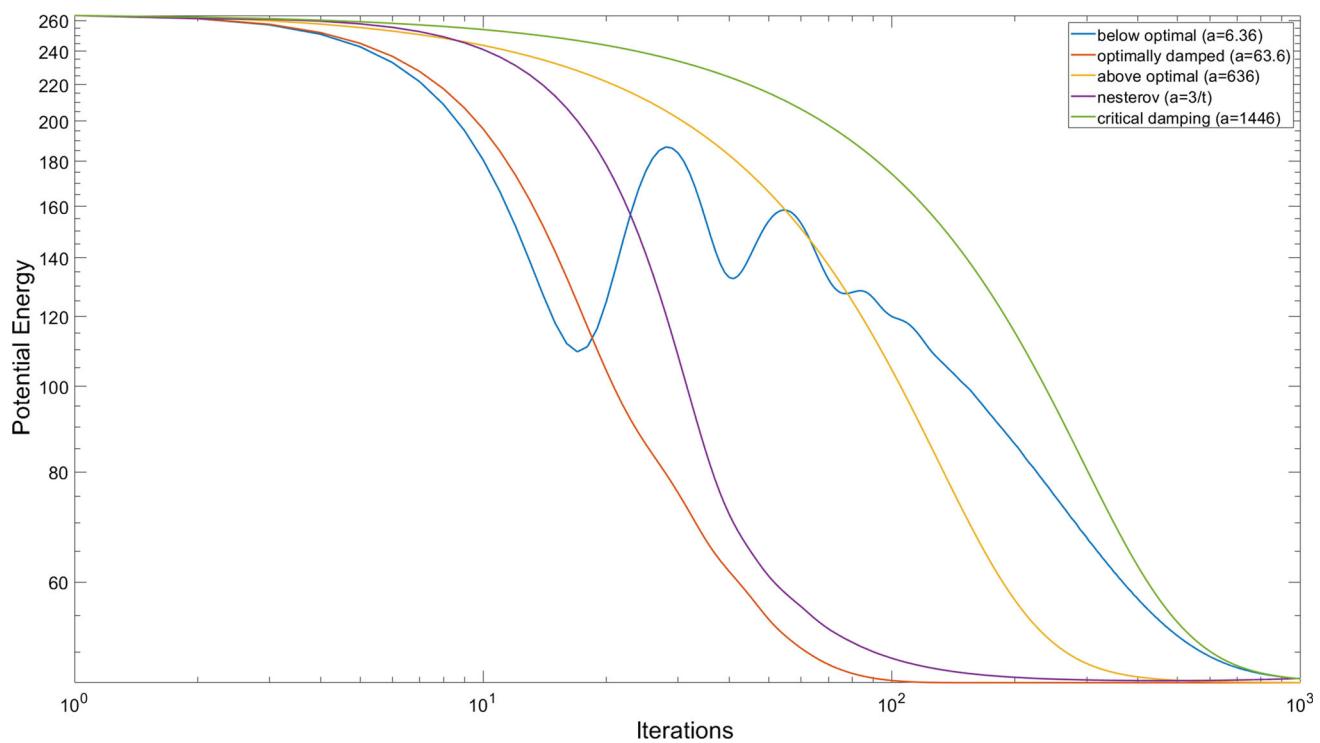
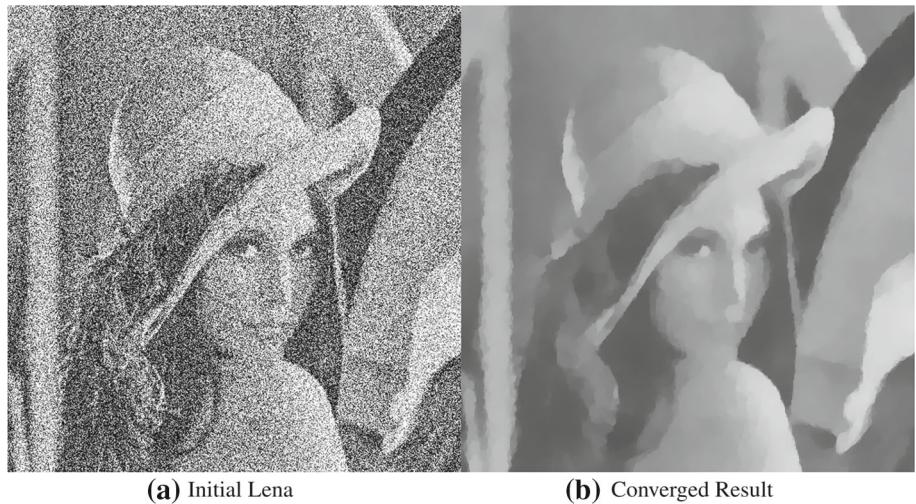
**Fig. 12** Comparison of PDE acceleration, primal–dual, and split Bregman for TV restoration of a noisy Lenna image with  $\lambda = 1000$ . Each algorithm was run for 150 iterations, which took 2.7 s for PDE acceleration, 3.3 s for primal–dual, and 28 s for split Bregman



**Fig. 13** Comparison of PDE acceleration, primal–dual, and split Bregman for TV restoration of a noisy Lenna image with  $\lambda = 7000$ . Each algorithm was run for 50 iterations, which took 0.85 s for PDE acceleration, 1.12 s for primal–dual, and 10.4 s for split Bregman



**Fig. 14** Initial (a) and final (b) condition of denoising experiment with variable damping. Note the final result does not change only the number of iterations required



**Fig. 15** Convergence experiment with variable damping for a Beltrami regularizer. Initial condition and converged result are given in Fig. 14

though, we find the flow does not fully converge, yet remains stable via the nonlinear effect discussed in Sect. 4.5, but instead tends to an oscillatory steady state. Figure 8 shows a snapshot of the steady state for various values of the time step  $\Delta t$ . For  $\Delta t \leq \Delta x$ , the steady state is a reasonable denoising; hence, we choose  $\Delta t = \Delta x$  or  $\Delta t = \Delta x/2$  in most of this paper. Note that this closely matches the suggested time step bound in (67) for a quantization level of 1/255, given the other parameters utilized here, which would come out to  $\Delta t \leq 1.189\Delta x$ .

Figures 9, 10, and 11 compare the energy decay against CPU time for denoising the Lena image with PDE acceleration, primal–dual, and split Bregman algorithms. The noise is additive zero mean Gaussian noise with standard deviation  $\sigma = 0.1$ , and the images take values in the interval  $[0, 1]$ . We note in Figs. 12 and 13 that PDE acceleration appears to yield a better quality image for the same energy level compared to primal–dual.

**Table 2** Coefficients and converge times of damping experiments for a 512 x 512 noisy Lena image

| Damping                       | Below optimal |            | Optimal |            | Above optimal |            | Nesterov |            | Critical |            |
|-------------------------------|---------------|------------|---------|------------|---------------|------------|----------|------------|----------|------------|
|                               | a             | Iterations | a       | Iterations | a             | Iterations | a        | Iterations | a        | Iterations |
| $\lambda = 1000, \beta^2 = 1$ | 6.36          | 1000       | 63.6    | 100        | 636           | 400        | 3/t      | 250        | 1446     | 1000       |

## 5.5 Variable Damping

We now give further considerations to variable damping. In Fig. 14, we show the initial condition and final converged result for five separate damping experiments. Note that the final result remains the same regardless of the damping coefficient that is chosen for a. From the linear analysis in (16) and our choice of  $\lambda = 1000$  and  $\beta = 1$  for the two tuning parameters, we have  $a = \sqrt{\lambda + \beta\pi^2}$  giving an optimal damping of 63.6. In Fig. 15, we compare an optimally damped system to a below optimal, above optimal, Nesterov, and critically damped system. The damping coefficients and convergence times in iterations are given in Table 2. The below optimal and above optimal are each one order of magnitude away from the optimal damping, respectively, and the critical damping is the point at which the second-order accelerated scheme is equivalent to gradient descent, i.e., the point at which the damping completely cancels out the momentum leaving only a first-order descent for the PDE. While the optimal damping will always give the fastest convergence for the PDE, if one is uncertain of the optimal damping, then using a greater than optimally damped but less than critically damped system will yield reasonable performance. While the Nesterov damping does converge faster than the above optimally damped example in Fig. 15, the increasing damping as a function of time will yield degraded performance and would likely necessitate an additional stopping criterion. Although the below optimally damped system is initially faster than the optimally damped system, it is subject to large oscillations in energy which while they do converge greatly slows down the final convergence time.

## 6 Conclusion

We employed the novel framework of *PDE acceleration*, based on momentum methods such as Nesterov and Polyak's heavy ball method, to calculus of variation problems defined for general functions on  $\mathbb{R}^n$ . The result was a very general set of *accelerated PDEs* whose simple discretizations efficiently solve the resulting class of optimization problems. We further analyzed their use in regularized inversion problems, where gradient descent diffusion equations get replaced by nonlinear wave equations within the framework of PDE

acceleration, with far more generous discrete time step conditions.

We presented results of experiments on image processing problems including Beltrami regularized denoising and inpainting, and total variation (TV) regularized denoising and deblurring. In all cases, we can achieve state-of-the-art results with very simple algorithms; indeed, the PDE acceleration update is a simple explicit forward Euler update of a nonlinear wave equation. Future work will focus on problems such as TV inpainting, where there is no fidelity, how to choose the damping parameter adaptively to further accelerate convergence, and applications to other problems in computer vision, such as Chan–Vese active contours [11].

## References

1. Attouch, H., Goudou, X., Redont, P.: The heavy ball with friction method, I. The continuous dynamical system: global exploration of the local minima of a real-valued function by asymptotic analysis of a dissipative dynamical system. *Commun. Contemp. Math.* **2**(01), 1–34 (2000)
2. Aubert, G., Kornprobst, P.: *Mathematical Problems in Image Processing: Partial Differential Equations and the Calculus of Variations*. Springer, New York (2006)
3. Bähr, M., Breuß, M., Wunderlich, R.: Fast explicit diffusion for long-time integration of parabolic problems. In: AIP Conference Proceedings, vol. 1863, p. 410002. AIP Publishing (2017)
4. Bottou, L.: Large-scale machine learning with stochastic gradient descent. In: Proceedings of COMPSTAT'2010, pp. 177–186. Springer (2010)
5. Boyd, S., Vandenberghe, L.: *Convex Optimization*. Cambridge University Press, Cambridge (2004)
6. Baravdish, G., Svensson, O., Gulliksson, M., Zhang, Y.: A damped flow for image denoising. arXiv preprint [arXiv:1806.06732](https://arxiv.org/abs/1806.06732) (2018)
7. Calatroni, L., Chambolle, A.: Backtracking strategies for accelerated descent methods with smooth composite objectives. arXiv preprint [arXiv:1709.09004](https://arxiv.org/abs/1709.09004) (2017)
8. Calder, J., Yezzi, A.: An accelerated PDE framework for efficient solutions of obstacle problems. Preprint (2018)
9. Chambolle, A., Pock, T.: A first-order primal–dual algorithm for convex problems with applications to imaging. *J. Math. Imaging Vis.* **40**(1), 120–145 (2011)
10. Chambolle, A., Pock, T.: An introduction to continuous optimization for imaging. *Acta Numer.* **25**, 161–319 (2016)
11. Chan, T.F., Vese, L.A.: Active contours without edges. *IEEE Trans. Image Process.* **10**(2), 266–277 (2001)
12. Goldstein, T., Osher, S.: The split Bregman method for  $\ell_1$ -regularized problems. *SIAM J. Imaging Sci.* **2**(2), 323–343 (2009)
13. Goudou, X., Munier, J.: The gradient and heavy ball with friction dynamical systems: the quasiconvex case. *Math. Program.* **116**(1–2), 173–191 (2009)

14. Hafner, D., Ochs, P., Weickert, J., Reißel, M., Grewenig, S.: FSI schemes: fast semi-iterative solvers for PDEs and optimisation methods. In: German Conference on Pattern Recognition, pp. 91–102. Springer (2016)
15. Kimmel, R., Malladi, R., Sochen, N.: Image processing via the beltrami operator. In: Asian Conference on Computer Vision, pp. 574–581. Springer (1998)
16. Nesterov, Y.: A method of solving a convex programming problem with convergence rate  $o(1/k^2)$ . Sov. Math. Dokl. **27**, 372–376 (1983)
17. Perona, P., Malik, J.: Scale-space and edge detection using anisotropic diffusion. IEEE Trans. Pattern Anal. Mach. Intell. **12**(7), 629–639 (1990)
18. Polyak, B.T.: Some methods of speeding up the convergence of iteration methods. USSR Comput. Math. Math. Phys. **4**(5), 1–17 (1964)
19. Ratner, V., Zeevi, Y.Y.: Image enhancement using elastic manifolds. In: 14th International Conference on Image Analysis and Processing (ICIAP 2007), pp. 769–774 (2007). <https://doi.org/10.1109/ICIAP.2007.4362869>
20. Rudin, L.I., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. Phys. D Nonlinear Phenom. **60**(1–4), 259–268 (1992)
21. Sochen, N., Kimmel, R., Malladi, R.: A general framework for low level vision. IEEE Trans. Image Process. **7**(3), 310–318 (1998)
22. Su, W., Boyd, S., Candes, E.: A differential equation for modeling Nesterov's accelerated gradient method: theory and insights. In: Advances in Neural Information Processing Systems, pp. 2510–2518 (2014)
23. Sundaramoorthi, G., Yezzi, A.: Accelerated optimization in the PDE framework: formulations for the manifold of diffeomorphisms. [arXiv:1804.02307](https://arxiv.org/abs/1804.02307) (2018)
24. Sundaramoorthi, G., Yezzi, A.: Variational PDE's for acceleration on manifolds and applications to diffeomorphisms. In: Neural Information Processing Systems (2018)
25. Sutskever, I., Martens, J., Dahl, G., Hinton, G.: On the importance of initialization and momentum in deep learning. In: International Conference on Machine Learning, pp. 1139–1147 (2013)
26. Trefethen, L.N.: Finite difference and spectral methods for ordinary and partial differential equations. unpublished text, available at <http://web.comlab.ox.ac.uk/oucl/work/nick.trefethen/pdetext.html> (1996)
27. Ward, C., Whitaker, N., Kevrekidis, I., Kevrekidis, P.: A toolkit for steady states of nonlinear wave equations: continuous time Nesterov and exponential time differencing schemes. [arXiv:1710.05047](https://arxiv.org/abs/1710.05047) (2017)
28. Weickert, J., Grewenig, S., Schroers, C., Bruhn, A.: Cyclic schemes for PDE-based image analysis. Int. J. Comput. Vis. **118**(3), 275–299 (2016)
29. Wibisono, A., Wilson, A.C., Jordan, M.I.: A variational perspective on accelerated methods in optimization. Proc. Natl. Acad. Sci. **113**(47), E7351–E7358 (2016)
30. Yezzi, A., Sundaramoorthi, G.: Accelerated optimization in the PDE framework: formulations for the active contour case. [arXiv:1711.09867](https://arxiv.org/abs/1711.09867) (2017)
31. Yezzi, A., Sundaramoorthi, G., Benyamin, M.: PDE acceleration for active contours. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2019)
32. Zosso, D., Bustin, A.: A primal-dual projected gradient algorithm for efficient beltrami regularization. UCLA CAM Report 14–52 (2014). <ftp://ftp.math.ucla.edu/pub/camreport/cam14-52.pdf>



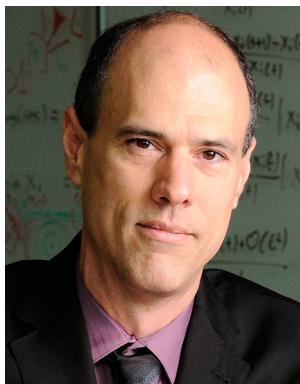
**Minas Benyamin** is an Electrical Engineering Ph.D. student at the Georgia Institute of Technology. He graduated with his bachelors in Electrical Engineering from the University of Maryland College Park in May 2016 and received his Masters in Electrical Engineering from the Georgia Institute of Technology in May 2018.



**Jeff Calder** is an assistant professor of mathematics at the University of Minnesota. He obtained his Ph.D. in Applied Mathematics from the University of Michigan in 2014, and a B.Sc. in Mathematics and Engineering and M.Sc. in Mathematics from Queen's University in 2008 and 2010, respectively. From 2014 to 2016, he was a Morrey Assistant Professor at the University of California, Berkeley, and he joined the faculty in the School of Mathematics at the University of Minnesota as an assistant professor in 2016. Dr. Calder's research interests lie at the intersection of partial differential equations (PDE), applied probability, and machine learning, and he is interested in applications in image processing and computer vision. Some central themes of his research involve PDE continuum limits for discrete problems in machine learning.



**Ganesh Sundaramoorthi** is currently Principal Research Scientist at United Technologies Research Center in East Hartford, CT, USA. His research throughout his career has been in computer vision and optimization, both in theory and applications. He has applied some of his research to develop technology for video analysis, seismic image analysis, electron microscopy images, and medical (MRI & CT) images. He was on the faculty of KAUST within the Electrical Engineering and Applied Mathematics and Computational Science departments between 2011 and 2018. He was a postdoctoral research associate at the University of California, Los Angeles from 2008 to 2011. His PhD is in Electrical and Computer Engineering from the Georgia Institute of Technology in Atlanta, GA, USA in 2008. His PhD developed shape optimization methods for computer vision that aided in technology for video tracking, and medical image analysis. His Bachelor's degrees were in Computer Engineering and Applied Mathematics, which he earned in 2003, also from Georgia Tech. He has served as Area Chair for the leading computer vision conferences, including IEEE Conference on Computer Vision and Pattern Recognition (CVPR) and IEEE International Conference on Computer Vision (ICCV).



**Anthony Yezzi** holds the position of Julian Hightower Chair Professor within the School of Electrical and Computer Engineering at Georgia Institute of Technology where he directs the Laboratory for Computational Computer Vision. He has over twenty years of research experience in shape optimization via geometric partial differential equations. He obtained his Ph.D. in Electrical Engineering in December 1997 from the University of Minnesota with a minor in mathematics.

After completing a postdoctoral research appointment at Massachusetts Institute of Technology, he joined the faculty at Georgia Tech in August 1999. Dr. Yezzi's research lies primarily within the fields of image processing and computer vision with a particular emphasis on medical imaging and 3D surface reconstruction. He has consulted for a number of companies including GE, 3M, MZA, Philips, Picker, and VTI. His work spans a wide range of image processing and vision problems including image denoising, edge-detection, segmentation, shape analysis, multi-frame stereo reconstruction, visual tracking, and registration. Some central themes of his research include curve and surface evolution, differential geometry, partial differential equations, and shape optimization.