

Sobolev Gradients and Image Interpolation*

Parimah Kazemi[†] and Ionut Danaila[‡]

Abstract. We present here a new image inpainting algorithm based on the Sobolev gradient method in conjunction with the Navier–Stokes model. The original model of Bertalmío, Bertozzi, and Sapiro [*Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2001, pp. 355–362] is reformulated as a variational principle based on the minimization of a well-chosen functional by a steepest descent method using Sobolev gradients. This new theoretical framework offers an alternative to the direct solving of a high-order PDE, with the practical advantage of an easier and more flexible computer implementation. In particular, the proposed algorithm does not require any constant tuning or advanced knowledge of numerical methods for Navier–Stokes equations (slope limiters, dynamic relaxation for Poisson equation, anisotropic diffusion steps, etc.). Using a straightforward finite difference implementation, we demonstrate, through various examples for image inpainting and image interpolation, that the novel algorithm is faster than the original implementation of the Navier–Stokes model, while providing results of similar quality. This paper also provides the mathematical theory for the analysis of the algorithm. Using an evolution equation in an infinite dimensional setting, we obtain global existence and uniqueness results as well as the existence of an ω -limit. This formalism is of more general interest and could be applied to other image processing models based on variational formulations.

Key words. image interpolation, Sobolev gradients, Navier–Stokes model, preconditioning

AMS subject classifications. 65M06, 65K10, 35Q30

DOI. 10.1137/11082453X

1. Introduction. Image inpainting refers to the process of filling in an occluded region in an image so that the seam between the original image and the inpainted region is undetectable by a typical viewer. Some examples of applications are restoring old photographs, removing unwanted objects from images such as text, or removing certain features of an image such as eye glasses. Image inpainting traditionally has been the work of professional artists. However, in recent years, much work has been done in the area of digital image inpainting. The idea of digital image inpainting is that the algorithm, using information from elsewhere in the digital image, fills in the unknown region so that a typical viewer cannot judge what parts of the image are from the original image and what parts are painted in. In general, the area to be altered consists of many pixels and complex geometries, and possibly spans the entire image. This establishes image inpainting as a challenging computational problem.

Currently there are a number of models and algorithms that are used in digital inpainting. The pioneering works in digital image inpainting include [7], [11], [17], and [19]. In [19], the

*Received by the editors February 14, 2011; accepted for publication (in revised form) February 6, 2012; published electronically May 1, 2012. This work was supported in part by the National Security Agency, contract TO-0129.

<http://www.siam.org/journals/siims/5-2/82453.html>

[†]3519 Roma Lane, Middleton, WI 53562 (parimah.kazemi@gmail.com).

[‡]Laboratoire de mathématiques Raphaël Salem, Université de Rouen, F-76801 Saint-Étienne-du-Rouvray, France (ionut.danaila@univ-rouen.fr).

authors present a method where the region to be inpainted is viewed as an occluded area. Then one connects T-junctions with the same grayscale values at the boundary using elastica minimizing curves. In [17], the authors extend this idea by using a variational formulation. They inpaint the region with unknown pixel values by connecting isophotes at the boundary using geodesic curves. In [11], the authors use a geometric model associated with Euler's elastica. The idea is to view the image as a surface and to minimize a linear combination of the curvature and arc length. More recently, the work in [8], [14], and [28] has provided state-of-the-art results in the area of digital image inpainting. In [12] and [6] a comprehensive survey of modern methods is given.

We focus here on the model proposed by Bertalmío, Bertozzi, and Sapiro (BBS), as presented in [5]. The model is based on a transport equation analogous to the transport of vorticity in an incompressible fluid. The resulting model is a Navier–Stokes-type time evolution. The approximated steady state solution of this model gives the corrected image. This model has several attractive features. It is able to fill in regions surrounded by different backgrounds as well as regions that cross through boundaries, it is capable of handling arbitrary topologies, and, as a third-order PDE, it forces continuity conditions on both the image intensity and the gradient of the intensity across the boundary. The main drawback of the method is that it requires a large number of iterations (several thousands) to converge, making it not very competitive against noniterative fast-marching methods (e.g., [8]).

In this paper, drawing motivation from the model presented in [5], we propose solving the third-order PDE by using a different formalism, based on the minimization of the corresponding least-squares derived functional. The idea is to use for the minimization a steepest descent method with Sobolev gradients, a method that we found effective in reducing the computational time for different applications (e.g., [13]). The novelty and the challenge of the present approach are that we deal with a third-order PDE, which is, in general, mathematically and computationally difficult to tackle. We establish in this contribution the mathematical framework for the analysis of the method and show that the numerical performance is considerably improved compared to the original implementation of the model described in [5]. We thus offer a new possibility for using the Navier–Stokes model as an alternative to fast-marching methods, which remain unbeatable with regard to computing performance, but are still technical since they need a careful tuning of constants or functional parameters (see also [6]).

In the method of calculus of variations, one often solves minimization problems by computing the Euler–Lagrange equations which are used as a direction of descent for the energy. Explicit iterative schemes using the Euler–Lagrange equations require many iterations to converge due to a tight restriction on the step size resulting from the CFL stability condition. In our minimization scheme, we compute a gradient with respect to a Sobolev metric. In practice, this amounts to preconditioning the Euler–Lagrange equations for use in minimization. The preconditioning in general allows for a larger time step when the resulting evolution equation is discretized in time. The Sobolev gradient method has proved to be very effective in a variety of image processing applications, for example, [9], [10], [21], and [23]. The focus of the current paper is to study the effect of preconditioning that is often associated with the Sobolev gradient, both in terms of the quality of the image and the computational efficiency. In contrast with the BBS model, we do not use a diffusion term, but instead apply a smoothing operator to the Navier–Stokes gradient flow. An advantage is that our evolution equation is well-posed

for all time, and we obtain that the minimizer is C^1 across the boundary and can be obtained as an ω -limit of the evolution equation. The minimization of the energy functional with the Sobolev gradient falls into the more general category of image regularization [3], [20], [24], [30], [29] since we expect the interpolated image to be smooth due to continuity properties of the gradient.

Although our results and applications are restricted to solving for the two-dimensional steady state arrived at by the Navier–Stokes model, our scheme is quite general. The ideas we present here can easily be adapted to computationally solve a vast variety of higher-order PDEs. Since we treat the problem as a variational problem, our scheme can also be applied to existing variational models in image processing (which almost always use the Euler–Lagrange equations for minimization). An interesting, although not immediate, further application would be the use of the Sobolev gradient method for variational problems such as those presented in [2]. Finally, our scheme is cast in both an infinite dimensional Hilbert space setting and analogously in a finite dimensional finite difference setting, making a strong connection between the theory of digital image analysis and applications.

The outline of this paper is as follows. In section 2, we give a brief explanation of the BBS model as presented in [5] and make a connection to the Navier–Stokes model for incompressible fluids. In section 3, we describe the variational formulation from which we obtain our gradient flow in a Hilbert space setting. We show that the flow has a unique global solution and an ω -limit at which minimization is achieved. In section 4 we obtain an expression for the Sobolev gradient. In section 5, we give the mathematical formulation in the discrete finite difference setting, and in section 6 we present the results from several examples which include the resulting images as well as a comparison of CPU time, time step size, and the effectiveness of our scheme in solving the boundary value PDE.

2. The Navier–Stokes model for image inpainting. Here we give a summary of the Navier–Stokes model for image inpainting as presented in [5], and also in [7], [4]. A digital grayscale image can be thought of as a set of m by n data points taking values between 0 and 255. The resulting function I defined on the m by n grid is called the image intensity. An isophote of I is a level-line in this context. The smoothness of the image is represented by the Laplacian of the intensity. The direction of the level-lines of the intensity I at the (i, j) th grid point is given by the direction that is perpendicular to the gradient of I at the point (i, j) .

The idea of the Navier–Stokes model for image inpainting is to propagate the gradient of the smoothness of the images in the direction of the isophotes. Thus one wants to iterate the evolution equation

$$(2.1) \quad I'(t) = \nabla^\perp I(t) \cdot \nabla \Delta I(t)$$

or perhaps, by adding an anisotropic diffusion term, iterate

$$(2.2) \quad I'(t) = \nabla^\perp I(t) \cdot \nabla \Delta I(t) + \nu \nabla \cdot (g |\nabla I(t)| \nabla I(t))$$

to a steady state solution

$$(2.3) \quad \nabla^\perp I \cdot \nabla \Delta I = 0.$$

In (2.2), a second term is added for its smoothing effects, where $\nu > 0$ is small and g is a diffusivity function.

The Navier–Stokes model for incompressible Newtonian fluids gives that the velocity field v and pressure p are coupled according to the equation

$$(2.4) \quad v'(t) + v(t) \cdot \nabla v(t) = -\nabla p(t) + \nu \Delta v(t) \text{ and } \nabla \cdot v(t) = 0.$$

In two dimensions, the divergence-free velocity field possesses a stream function ψ so that $\nabla^\perp \psi = v$. The vorticity can be expressed as $w = \Delta \psi$ and satisfies the advection diffusion equation

$$(2.5) \quad w'(t) = -v(t) \cdot \nabla w(t)$$

when the viscosity ν is zero. The steady state must satisfy

$$(2.6) \quad \nabla^\perp \psi \cdot \nabla \Delta \psi = 0.$$

This is the incompressible Euler equation for fluid flow. We refer the reader to [26] for background on the theory of the Navier–Stokes equation.

Another way to say this is that the gradient of the stream function ψ and the gradient of the Laplacian of the stream function must be parallel. Thus the analogy between the Navier–Stokes model and image inpainting is that the image intensity acts as the stream function in the Navier–Stokes model. In our approach, we consider an alternate to the method of Bertalmio as presented in [5] and minimize the norm of the left-hand side using an evolution equation based on a Sobolev gradient method.

3. A variational formulation in the continuous setting. In this section, we describe our scheme in a Hilbert space setting. The main idea is that we form an energy functional by taking the norm of the left-hand side of (2.3). We then search for a minimum of this energy using a gradient flow with a Sobolev gradient. We start by stating the definitions and results that we will need from the theory of Sobolev spaces. We then define the variational problem and study properties of the resulting functional. We argue that every critical point must be a solution of (2.3). We define the Sobolev gradient in a general setting, and, incorporating the boundary value into the gradient, we obtain a gradient flow that has a unique global solution. We obtain a critical point of the energy as an ω -limit of the gradient flow. Since a critical point of the energy corresponds to a zero of the energy, we obtain that our energy functional does indeed possess a minimum. We discuss regularity properties of the minimizer.

3.1. Sobolev spaces. We review here the basic information regarding the theory of Sobolev spaces. The information provided here, as well as the notation, is taken from [1]. Let Ω be a bounded open subset of \mathbb{R}^2 . Recall that

$$W^{m,p}(\Omega) \equiv \{u \in L^p(\Omega) : D^\alpha u \in L^p(\Omega) \text{ for } 0 \leq |\alpha| \leq m\},$$

where for a multi-index α , $D^\alpha u$ denotes the α th partial derivative of u taken in the distributional sense. Recall also that

$$H^{m,p}(\Omega) \equiv \text{the completion of } \{u \in C^m(\Omega) : \|u\|_{m,p} < \infty\},$$

where $C^m(\Omega)$ is the space of m times continuously differentiable functions and

$$\|u\|_{m,p} = \left(\sum_{0 \leq |\alpha| \leq m} \|D^\alpha u\|_p^p \right)^{1/p}$$

with $\|\cdot\|_p$ denoting the L^p norm. $H_0^{m,p}(\Omega)$ is the closure of the infinitely differentiable functions with compact support in Ω in $W^{m,p}(\Omega)$. It is a result (Theorem 5.37 of [1]) that under sufficient regularity conditions on the boundary of Ω , the trace of $D^\alpha u$ is zero for all $u \in H_0^{m,p}(\Omega)$ and $|\alpha| \leq m$.

It follows that for every open domain, $H^{m,p}(\Omega) = W^{m,p}(\Omega)$. Since we wish to work in a Hilbert space setting, we fix $p = 2$ and refer to the space $H^{m,2}(\Omega)$ as H^m , $H_0^m = H_0^{m,2}(\Omega)$, and $L^2(\Omega) = L$.

Let $Du = \{D^\alpha u : 1 \leq |\alpha| \leq k\}$. We can see that $\left\{ \begin{pmatrix} u \\ Du \end{pmatrix} : u \in H^m \right\}$ is a closed subspace of $L(\Omega^m)$, where $\Omega^m = \cup_{|\alpha| \leq m} \Omega_\alpha$. Thus there exists a unique orthogonal projection $P = P(m)$ from $L(\Omega^m)$ onto $\left\{ \begin{pmatrix} u \\ Du \end{pmatrix} : u \in H^m \right\}$. We mention this here as we will need this projection when constructing the Sobolev gradient in section 4.2. We also note that H_0^m is a closed subspace of H^m . Thus there exists a unique orthogonal projection of H^m onto H_0^m which we denote by $P_0 = P_0(m)$.

Suppose that Ω is bounded and has C^m boundary $\partial\Omega$. It is a result (see [1]) that for $u \in H^m$, the boundary traces of u can be defined as $f_\alpha(u) = D_\alpha u|_{\partial\Omega}$ for $0 \leq |\alpha| \leq m$. Further, $f_\alpha(u)$ is in $L^2(\partial\Omega)$. If $u \in H_0^m$, then $f_\alpha(u) = 0$ for all α . Finally, we remind the reader of the part of the Rellich–Kondrachov theorem that we will need.

Theorem 3.1. *Suppose that $\Omega \subset \mathbb{R}^2$ is a bounded domain with a smooth boundary. The embeddings*

$$H^{j+m}(\Omega) \rightarrow C^j(\bar{\Omega})$$

and

$$H^{j+m} \rightarrow H^j$$

are compact when $m \geq 2$.

Here $C^j(\bar{\Omega})$ denotes the j times continuously differentiable functions on $\bar{\Omega}$.

3.2. Defining the minimization problem. Assume that Ω is a bounded open set in \mathbb{R}^2 and has smooth boundary. For $u \in H^3$, define F so that

$$(3.1) \quad F(Du) = \nabla^\perp u \cdot \nabla \Delta u.$$

Now if $u \in H^3$, then $D^\alpha u \in H^2$ when $|\alpha| = 1$. Since H^2 is compactly embedded in $C(\bar{\Omega})$, for $u \in H^3$, ∇u is a bounded continuous function, and $|\nabla u| \leq c\|u\|_{H^3}$ for some constant c independent of u . $|\cdot|$ denotes the sup norm here, and in the rest of the paper, $\|\cdot\|$ denotes the L^2 norm, and $\|\cdot\|_{H^m}$ the H^m norm. Thus

$$(3.2) \quad \|F(Du)\| = \left(\int_\Omega |\nabla^\perp u \cdot \nabla \Delta u|^2 \right)^{1/2} \leq |\nabla u| \|\nabla \Delta u\| \leq c \|u\|_{H^3}^2,$$

and we see that for $u \in H^3$, $F(Du) \in L^2$ and thus the energy

$$(3.3) \quad E(u) = \frac{\|F(D(u_0))\|^2}{2}$$

is well defined. To reduce notation, we write $u_0 = P_0u$, where P_0 is the orthogonal projection of H^3 onto H_0^3 .

In the model of BBS, $u \in H^3$ is sought so that $F(Du_0) = 0$ and $u = g$ on the boundary of Ω for some predefined function g . We assume that g is C^1 on the boundary of Ω .

First we show that E is C^2 differentiable and then explain why a minimizer of E would correspond to a zero of $F \circ D$.

Theorem 3.2. $E : H^3 \rightarrow \mathbb{R}$ as defined in (3.3) is C^2 Fréchet differentiable.

We compute here the first and second Fréchet derivatives of E . We regard the first Fréchet derivative of E at $u \in H^3$ as a bounded linear operator on H^3 and the second Fréchet derivative of E at $u \in H^3$ as a bounded and symmetric bilinear operator on H^3 .

Proof. We compute a Fréchet derivative of F as given in (3.1) to obtain

$$(3.4) \quad F'(Du)Dh = \nabla^\perp h \cdot \nabla \Delta u + \nabla^\perp u \cdot \nabla \Delta h.$$

One can also compute $F''(Du)(Dh, Dk)$ in order to see that $F \circ D$ is C^2 Fréchet differentiable on H^3 . Next we can compute the first and second Fréchet derivatives of E to see that

$$(3.5) \quad E'(u)h = \langle F'(Du_0)Dh_0, F(Du_0) \rangle$$

and

$$E''(u)(h, k) = \langle F'(Du_0)Dh_0, F'(Du_0)Dk_0 \rangle + \langle F''(Du_0)(Dh_0, Dk_0), F(Du_0) \rangle.$$

E being C^2 Fréchet differentiable follows from the differentiability of F .

Notice from (3.4) that $F'(Du)Du = 2F(Du)$. From (3.5), we see that

$$(3.6) \quad E'(u)u_0 = \langle F'(Du_0)Du_0, F(Du_0) \rangle = 2\|F(Du_0)\|^2 = 4E(u). \quad \blacksquare$$

Notice also from (3.6) that $E'(u)u_0 = 2\|F(Du_0)\|^2$. Hence, in order to solve the boundary value problem, it suffices to find $u \in H^3$ so that $u = g$ on $\partial\Omega$ and $E'(u)h = 0$ for all $h \in C_0^\infty(\Omega)$.

3.3. The Sobolev gradient. The idea of Sobolev gradients [18] is that, given a C^1 function ϕ that is everywhere defined on a Hilbert space H , we can represent the Fréchet derivative of E at u using a member of the Hilbert space. This is due to the Riesz representation theorem because if ϕ is C^1 , then $\phi'(u)$ is a bounded linear functional on H . Thus there exists a unique element of H , which we denote by $\nabla E(u)$, so that

$$(3.7) \quad \phi'(u)h = \langle h, \nabla \phi(u) \rangle_H \quad \forall h \in H.$$

One then considers the gradient flow

$$(3.8) \quad z(0) = x_0 \in H \text{ and } z'(t) = -\nabla \phi(z(t)).$$

It follows that z defines a path along which the energy is decreasing. Assuming one has existence of such a z , then it is important to study the asymptotic limit as it is in this limit that we hope to achieve a critical point.

For this problem we consider a gradient with respect to the H^3 inner product. In order to obtain an ω -limit, however, we need to map this gradient into a higher order space. In this section, we give two results from [18] that we will modify for this work and define our evolution equation.

Theorem 3.3. *Suppose ϕ is a nonnegative valued C^1 function on a Hilbert space H . Define the gradient of ϕ at u as in (3.7). If $\nabla\phi$ is a locally Lipschitz function from H to H , then for $x_0 \in H$ the gradient flow (3.8) has a unique solution for all $t \geq 0$.*

Theorem 3.4. *Suppose, under the conditions of Theorem 3.3, that z is a global solution of (3.8) with energy ϕ . Further suppose that $\phi'(u)u \geq 0$ for all $u \in H$; then the range of z is a bounded subset of H .*

The main idea of this theorem is that

$$(\|z\|_H^2)'(t) = \langle z(t), z'(t) \rangle_H = -\langle z(t), \nabla_H \phi(z(t)) \rangle_H = -\phi'(z(t))z(t) \leq 0$$

by assumption. This implies that $t \rightarrow \|z(t)\|_H^2$ is nonincreasing and thus the range of z is bounded in H .

Since E , as defined in (3.3), is Fréchet differentiable, $E'(u)$ is a bounded linear functional on H^3 . Thus for each $u \in H^3$, there exists a unique member of H^3 so that

$$E'(u)h = \langle h, \nabla_{H^3} E(u) \rangle_{H^3}.$$

For $x \in H_0^3$ and $k > 3$, consider the mapping from H_0^k to \mathbb{R} given by

$$y \rightarrow \langle y, x \rangle_{H^3}.$$

One can see that this is a bounded linear mapping. Thus there exists a unique element $M_{k,0} x \in H_0^k$ so that

$$\langle y, x \rangle_{H^3} = \langle y, M_{k,0} x \rangle_{H^k}$$

for all $y \in H_0^k$. Now consider the operator $M_{k,0}$. In [16], we obtained that this operator is bounded from H_0^3 to H_0^k with norm less than or equal to one.

We define the gradient flow

$$(3.9) \quad z(0) = u_0 \text{ and } z'(t) = -M_{k,0} P_0 \nabla_{H^3} E(z(t)).$$

where $u_0 \in H^k$ and $u_0 = g$ on $\partial\Omega$. Here P_0 is the projection of H^3 onto H_0^3 .

3.4. Global existence, uniqueness, and asymptotic convergence. In this section we give results regarding the global existence and uniqueness of the gradient flow (3.9). We show that a unique solution exists for this system and that the range of z is bounded in H^k . This allows us to extract a subsequence from the range of z which converges to a zero of the gradient, thus giving the existence of a stationary point and a solution to the minimization problem. We closely follow the developments in Chapter 4 of [18].

Theorem 3.5. *The gradient flow in (3.9) has a unique global solution $z \in C^1([0, \infty), H^k)$.*

We follow the proof of Theorem 3.3 presented in [18] with only a slight modification to account for the presence of $M_{k,0}P_0$ in our gradient flow.

Proof. In order to show that the gradient flow given in (3.9) has a local solution, we need to show that the gradient is locally Lipschitz from H^k to H^k . We showed in Theorem 3.2 that E is C^2 on H^3 . Note that for any C^2 function ϕ defined on a Hilbert space H , with gradient as in (3.7), there exist a constant c_u and a ball B about u so that if $v \in B$, then

$$|\langle h, \nabla\phi(u) - \nabla\phi(v) \rangle_H| = |(\phi'(u) - \phi'(v))h| \leq c_u|u - v|_H|h|_H.$$

Thus if we take $h = \nabla\phi(u) - \nabla\phi(v)$, then we see that

$$|\nabla\phi(u) - \nabla\phi(v)|_H \leq c_u|u - v|_H.$$

Hence if ϕ is a C^2 function defined on a Hilbert space, then the gradient of ϕ as defined in (3.7) is locally Lipschitz. From this we conclude that $\nabla_3 E : H^3 \rightarrow H^3$ is locally Lipschitz. $M_{k,0}P_0\nabla_{H^3}E$ is locally Lipschitz from H^k to H^k as for $u, v \in H^k$,

$$\begin{aligned} \|M_{k,0}P_0(\nabla_{H^3}E(u) - \nabla_{H^3}E(v))\|_{H^k} &\leq \|P_0(\nabla_{H^3}E(u) - \nabla_{H^3}E(v))\|_{H^3} \\ &\leq \|(\nabla_{H^3}E(u) - \nabla_{H^3}E(v))\|_{H^3} \leq c_u\|u - v\|_{H^3} \leq c_u\|u - v\|_{H^k}. \end{aligned}$$

We conclude that the system (3.9) has a local solution.

Let T be the largest number so that a solution exists on $[0, T)$. In order to show that the flow has a global solution, it suffices to show that $\lim_{t \rightarrow T^-} z(t)$ exists. This holds true, provided that there is a constant m so that

$$(3.10) \quad \|z(a) - z(b)\|_{H^k} \leq m$$

for all $0 < a, b < T$. Recalling that P_0 is symmetric with respect to $\langle \cdot, \cdot \rangle_{H^3}$ and $z'(t) \in H_0^k$ and hence in the range of P_0 for all t , we have that

$$\begin{aligned} \|z(a) - z(b)\|_{H^k}^2 &\leq \left(\int_a^b \|z'\|_{H^k} \right)^2 \leq \text{(using Cauchy-Schwarz)} \\ (b-a) \int_a^b \|z'\|_{H^k}^2 &= (b-a) \int_a^b \langle z', z' \rangle_{H^k} = -(b-a) \int_a^b \langle z', M_{k,0}P_0\nabla_{H^3}E(z) \rangle_{H^k} \\ &\leq -T \int_a^b \langle z', P_0\nabla_{H^3}E(z(t)) \rangle_{H^3} = -T \int_a^b \langle P_0z', \nabla_{H^3}E(z(t)) \rangle_{H^3} \\ &= -T \int_a^b \langle z', \nabla_{H^3}E(z(t)) \rangle_{H^3} = -T \int_a^b E'(z(t))z'(t) = -T \int_a^b (E \circ z)' \leq TE(z(0)). \end{aligned}$$

Thus (3.10) holds true, and we have that the flow (3.9) has a global solution. Uniqueness follows from the fact that the projected gradient is a locally Lipschitz function and from the fundamental theory for existence and uniqueness of ODEs. ■

We show next that the flow (3.9) has an ω -limit which is a minimizer of the energy and a solution to the boundary value problem.

Lemma 3.6. *There is an unbounded sequence $\{t_n\}_{n \geq 1}$ so that $M_{k,0}P_0 \nabla_{H^k} E(z(t_n))$ converges to zero in H^k .*

Proof. Using the same analysis as in the previous proof, we note that

$$\int_0^\infty \|M_{k,0}P_0 \nabla_{H^3} E(z)\|_{H^k}^2 \leq - \int_0^\infty (E \circ z)' \leq E(z(0)).$$

From this it follows that

$$\int_0^\infty \|M_{k,0}P_0 \nabla_{H^3} E(z)\|_{H^k}^2$$

is bounded, and hence the conclusion follows. ■

Lemma 3.7. *The range of z is bounded as a subset of H^k .*

The proof is similar to the proof of Theorem 3.4 given in [18] with only a slight modification.

Proof. Let $h(t) = .5\|z(t)\|_{H^k}^2$, and note that

$$\begin{aligned} h'(t) &= \langle z(t), z'(t) \rangle_{H^k} = -\langle z(t), M_{k,0}P_0 \nabla_{H^3} E(z(t)) \rangle_{H^k} \\ &= -\langle P_0 z(t), \nabla_{H^3} E(z(t)) \rangle_{H^3} = -E'(z(t))P_0 z(t) = -4E(z(t)), \end{aligned}$$

the last equality following from (3.6). Thus h' is never positive, and hence h is bounded. ■

Theorem 3.8. *Suppose $k \geq 5$ in (3.9). Then there exists a sequence of unbounded numbers $\{t_n\}_{n \geq 1}$ so that $\{z(t_n)\}_{n \geq 1}$ converges in H^3 to $u \in C^1(\bar{\Omega})$ and $E'(u)h = 0$ for all $h \in C_0^\infty(\Omega)$.*

Proof. Since the range of z is bounded in H^k for $k \geq 3$, using the compact embedding of H^k into H^3 , there exists a sequence $\{t_n\}_{n \geq 1}$ so that $\{z(t_n)\}_{n \geq 1}$ converges to $u \in H^3$.

Using Theorem 3.6, we can assume $\{t_n\}_{n \geq 1}$ is chosen so that $\{M_{k,0}P_0 \nabla_{H^3} E(z(t_n))\}_{n \geq 1}$ converges to zero in H^k . Using the regularity properties that are derived in Theorem 3.5, $M_{k,0}P_0 \nabla_{H^3} E(z(t_n))$ converges to 0 in H^k . For $h \in C_0^\infty$,

$$\begin{aligned} E'(u)h &= \langle h, \nabla_{H^3} E(u) \rangle_{H^3} \\ &= \langle P_0 h, \nabla_{H^3} E(u) \rangle_{H^3} = \langle h, P_0 \nabla_{H^3} E(u) \rangle_{H^3} \\ &= \langle h, M_{k,0}P_0 \nabla_{H^3} E(u) \rangle_{H^k} = 0. \end{aligned}$$

Since H^3 is embedded in $C^1(\bar{\Omega})$, $u \in C^1(\bar{\Omega})$. ■

We also note that $z(t)$ agrees with g on the boundary of Ω for all t . Recall that z has range in H^k , which is compactly embedded in $C^1(\bar{\Omega})$. Let $f(t) = z(t)|_{\partial\Omega}$. Then f is C^1 , and $|f'(t)| = |M_{k,0}P_0 \nabla_{H^3} E(z(t))|_{\partial\Omega} = 0$. Since

$$|f(t) - g| = |f(t) - f(0)| \leq \int_0^t |f'| = 0,$$

the assertion follows. The norm $|\cdot|$ here indicates the sup norm.

Regarding the uniqueness of the solution of the boundary value problem

$$\nabla^\perp u \cdot \nabla \Delta u = 0 \text{ in } \Omega \text{ and } u = g \text{ on } \partial\Omega,$$

we refer the reader to [5], where an example is given to illustrate that the solution of the boundary value problem is not unique. It is also argued that some level of nonuniqueness is required for the inpainting problem as one would like to be able to “choose” the best interpolation for the inpainting problem.

4. An expression for the gradient. Here we give an expression for the Sobolev gradient. We first derive the Euler–Lagrange equation for our energy functional. We observe that the Euler–Lagrange equation can be viewed as a gradient of the energy obtained with respect to an L^2 inner product. We then obtain an expression for the Sobolev gradient using the projection P discussed in section 3.1 and compare the Sobolev gradient with the Euler–Lagrange equation, making a key distinction between the two by comparing the regularity properties of the resulting gradients.

4.1. The Euler–Lagrange equation for E . Recall the expression for $E'(u)$ as given by (3.5). In order to obtain the Euler–Lagrange equation, first we need to find some vector \vec{v} whose components are in L^2 so that

$$(4.1) \quad \langle F'(Du)Dh, F(Du) \rangle = \langle Dh, \vec{v} \rangle.$$

For this purpose, let $y \in L^2$ and recall the expression for $F'(Du)Dh$ as given by (3.4) to obtain

$$\langle F'(Du)Dh, y \rangle = \langle \nabla^\perp h \cdot \nabla \Delta u + \nabla^\perp u \cdot \nabla \Delta h, y \rangle.$$

Since $\nabla^\perp h \cdot \nabla \Delta u = -\nabla h \cdot \nabla^\perp \Delta u$, we have that

$$\langle F'(Du)Dh, y \rangle = \langle \nabla h, -y \cdot \nabla^\perp \Delta u \rangle + \langle \nabla \Delta h, y \cdot \nabla^\perp u \rangle.$$

Here, when c is a scalar, $c \cdot \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} cx \\ cy \end{pmatrix}$.

From this calculation we have that the nonzero components of \vec{v} are $v_{\alpha_1} = -F(Du) \cdot \nabla^\perp \Delta u$ and $v_{\alpha_2} = F(Du) \cdot \nabla^\perp u$ so that

$$\langle Dh, F(Du) \rangle = \langle \nabla h, v_{\alpha_1} \rangle + \langle \nabla \Delta h, v_{\alpha_2} \rangle.$$

With this notation, under further regularity assumptions on u , the Euler–Lagrange equation for E becomes

$$(4.2) \quad \nabla_{EL} E(u) = D^* \vec{v},$$

where D^* denotes the adjoint of D as a closed and densely defined operator. Note that for all $h \in L^2$,

$$E'(u)h = \langle h, \nabla_{EL} E(u) \rangle.$$

Thus the Euler–Lagrange equation, when defined, can be viewed as an L^2 gradient for E .

4.2. An orthogonal projection. Recall from section 3.1 that P is the orthogonal projection of $L^2(\Omega^k)$ onto $\left\{ \begin{pmatrix} h \\ Dh \end{pmatrix} : h \in H^k \right\}$ with Dh denoting all partial derivatives of h up to order k . Recall also the expression for $E'(u)$ as given by (3.4) and (3.5). Then we have the following:

$$\begin{aligned} E'(u)h &= \langle F'(Du)Dh, F(Du) \rangle \\ &= \langle Dh, \vec{v} \rangle = \left\langle \begin{pmatrix} h \\ Dh \end{pmatrix}, \begin{pmatrix} 0 \\ \vec{v} \end{pmatrix} \right\rangle \\ &= \left\langle P \begin{pmatrix} h \\ Dh \end{pmatrix}, \begin{pmatrix} 0 \\ \vec{v} \end{pmatrix} \right\rangle \\ &= \left\langle Dh, P \begin{pmatrix} 0 \\ \vec{v} \end{pmatrix} \right\rangle = \left\langle h, \Pi P \begin{pmatrix} 0 \\ \vec{v} \end{pmatrix} \right\rangle_{H^k}, \end{aligned}$$

where $\Pi \begin{pmatrix} x \\ y \end{pmatrix} = x$. An expression for P , given in [18], is

$$P = \begin{pmatrix} (I + D^*D)^{-1} & D^*(I + DD^*)^{-1} \\ D(I + D^*D)^{-1} & I - (I + DD^*)^{-1} \end{pmatrix},$$

where D^* denotes the adjoint of D when viewed as a closed and densely defined linear operator on $L^2(\Omega)$. From this formula we see that

$$(4.3) \quad \nabla_{H^k} E(u) = P \begin{pmatrix} 0 \\ \vec{v} \end{pmatrix} = D^*(I + DD^*)^{-1} \vec{v}.$$

We note that if u satisfies additional regularity so that \vec{v} is in the domain of D^* , then

$$(4.4) \quad D^*(I + DD^*)^{-1} \vec{v} = (I + D^*D)^{-1} D^* \vec{v}.$$

Thus the Sobolev gradient can be viewed as a smoother gradient than the Euler–Lagrange equation which is the descent direction used in most optimization problems. In case $k = 1$, $(I + D^*D)^{-1}$ is the resolvent of the Neumann Laplacian $(I - \Delta)^{-1}$. For $k > 1$, a nice argument is given in [10] to show that $(I + D^*D)^{-1}$ is equivalent to $(I - \Delta)^{-k}$. Thus we see that using a gradient in a higher order Sobolev space is equivalent to preconditioning the Euler–Lagrange equations using the resolvent of the Neumann Laplacian as the preconditioner. A detailed discussion on the preconditioning resulting from the use of gradients in high-order Sobolev spaces is presented in the appendix.

5. Finite dimensional implementation. We discretize, using finite differences, (4.3) or, equivalently, (4.4) since in a finite dimension space the two expressions are equal. Here u denotes the image intensity, a real valued function defined on a rectangular domain R of \mathbb{R}^2 .

5.1. The inpainting domain. The inpainting domain is specified by the user. This can be done by marking the regions to be inpainted using photo editing software such as Gimp or Paint. We then determine the inpainting region, Ω , and the inpainting region plus its boundary, Ω' . We consider $u(i, j)$ to be in Ω' if $u(k, l) \in \Omega$ for some (k, l) with $0 \leq |i - k| + |j - l| \leq 2$. We label u_0 to be u restricted to Ω and u' to be u restricted to Ω' . u_0 and u' are converted into vectors using the natural ordering $(i + 1, j) > (i, j)$ and then $(i, j + 1) > (i, j)$.

5.2. The finite difference operators D_1 , D_2 , and Δ . Let D_1 , D_2 , and Δ be matrices defined on the interior of R so that

$$\begin{aligned} D_1 u_{i,j} &= \frac{1}{2}(u_{i+1,j} - u_{i-1,j}), \\ D_2 u_{i,j} &= \frac{1}{2}(u_{i,j+1} - u_{i,j-1}), \text{ and} \\ \Delta u_{i,j} &= (u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1} - 4u_{i,j}) \end{aligned}$$

for all $(i, j) \in \Omega$. We use second-order centered differences to approximate the first partial derivatives and the five point discretization of the Laplacian. Define also $D_1 \Delta = D_1 * \Delta$ and $D_2 \Delta = D_2 * \Delta$. In order to enforce the boundary conditions, we eliminate the rows of each of these operators corresponding to pairs $(i, j) \in R - \Omega$ and the columns corresponding to pairs $(i, j) \in R - \Omega'$. Finally we define $(I - \Delta)^{-1} : \Omega \rightarrow \Omega$ to be the resolvent of the Laplacian.

5.3. Discretizing the gradient. Here we discretize the Sobolev gradient. Note that in the finite dimensional case, (4.4) holds true. First we discretize $F(Du)$ as

$$(5.1) \quad F(Du) = D_2 u' * D_1 \Delta u' - D_1 u' * D_2 \Delta u',$$

where for two vectors x and y , $x * y$ denotes the vector consisting of their elementwise product. Next we discretize the Euler–Lagrange equation as follows. Discretize \vec{v} given in (4.2) so that

$$\begin{aligned} v_{\alpha_1} &= -F(Du) * \begin{pmatrix} -D_2 \Delta u' \\ D_1 \Delta u' \end{pmatrix}, \\ v_{\alpha_2} &= F(Du) * \begin{pmatrix} -D_2 u' \\ D_1 u' \end{pmatrix}. \end{aligned}$$

Then the discretized Euler–Lagrange equation for this problem becomes

$$(5.2) \quad g_{EL} = D_1' v_{\alpha_1}(1) + D_2' v_{\alpha_1}(2) + \Delta(D_1' v_{\alpha_2}(1) + D_2' v_{\alpha_2}(2)),$$

where D_i' denotes the adjoint of D_i .

5.4. Smoothing the Euler–Lagrange equation. In order to use the Sobolev gradient as given in (4.4), we need to discretize the operator $(I + D^*D)$. Once we have a discretization of this operator, we obtain the discrete Sobolev gradient by solving the equation

$$(I + D^*D)g_{Sob} = g_{EL}.$$

Since D involves partial derivatives of order up to three, a discretization of this operator would force us to solve a very complex linear system, which can be computationally very time consuming. A better approach is to obtain an equivalent operator to use for smoothing the Euler–Lagrange equation. A good choice turns out to be $(I - \Delta)^3$. A good explanation for this is given in [10]. Thus in order to solve for the Sobolev gradient, we solve the linear system

$$(5.3) \quad (I - \Delta)^3 g_{Sob} = g_{EL}.$$

This is computationally the most expensive part of the algorithm. To solve the linear system, a Cholesky factorization of $(I - \Delta)$ is performed to obtain an upper and lower triangular matrix. Then the linear system

$$(I - \Delta)x = y$$

is solved three times using the factorization in order to apply the smoothing operator three times. This is a convenient method since homogeneous Dirichlet boundary conditions naturally apply at each step of the factorization.

5.5. Discretization in time. In order to discretize the system (3.9) in time, we use an explicit Euler scheme with a locally minimizing time step. Thus we generate a sequence of images according to

$$u_0(n+1) = u_0(n) - t_n g_{Sob}(n).$$

We remark that several options are available for time discretization. One can choose between an explicit or an implicit scheme. One can also use a fixed time step. For our purpose, we chose the explicit scheme since computing the gradient here is more expensive than computing the value of the functional. Thus an implicit scheme that would require us to compute the gradient several times at each step is a lot more expensive than an explicit scheme with a line search algorithm to locally compute the optimum time step.

6. Results. We present here the results from our simulations. The implementation of the discrete operators derived in the previous section using finite differences is quite straightforward if a language with vector programming capabilities is used. The algorithm is programmed in MATLAB, and all operations are written in matrix form. We run our simulations on an IBM laptop with 1.5 GB RAM and 1.4GHz Pentium M processor. In each case the starting point for the simulation was the solution of Laplace's equation

$$(6.1) \quad \Delta u = 0 \text{ and } u|_{\partial\Omega} = g,$$

which was solved by using a successive overrelaxation method. The boundary condition function g takes values from the neighboring nonpainted domain. In the examples we present, we normalize all images so that the maximum pixel value is one. This choice for the initial guess of the solution looks natural if the Navier–Stokes analogy is used: (6.1) then represents an irrotational fluid field, i.e., with zero vorticity $\omega = \Delta\psi = 0$, which will be iterated to find a nonzero vorticity (source) term that is present in the model. In the method of BBS, the initial condition is set after performing several steps of anisotropic diffusion following (2.2).

For comparison purposes, we also implemented the method of BBS as described in [7]. We incorporated slope limiters for the convection term to propagate fronts with sharp variation and every 50 time steps applied a few iterations of the Perona–Malik scheme for anisotropic diffusion to avoid spurious oscillations:

$$(6.2) \quad I'(t) = \nabla \cdot c(x, y, t) \nabla I(t), \text{ where } c(x, y, t) = e^{-\|\nabla I\|/k}.$$

As many of the specific details such as the term used for anisotropic diffusion are not available in the original paper of BBS, an exact reproduction of their results proved to be difficult. As a consequence, the comparison presented in the following should be regarded with caution. Nevertheless, reported convergence properties of the method of BBS are recovered by our implementation.

6.1. Comparison with the original implementation of the Navier–Stokes model by Bertalmío, Bertozzi, and Sapiro. We perform a full quantitative comparison displaying convergence curves using our minimization procedure and the Navier–Stokes model of BBS. The original model is proposed for solving the equation $\nabla^\perp u \cdot \nabla \Delta u = 0$ by evolving this PDE in time adding anisotropic diffusion. Typical methods coming from fluid dynamics literature are used, and therefore, several delicate and technical choices are necessary: the choice of the limiters for the discretization of convection terms, the relaxed Laplacian solvers for the vorticity needing the tuning of a constant, the choice of how often and under which form to introduce the diffusion term to avoid spurious oscillations. Besides, no convergence analysis is provided to state that a solution of the PDE is obtained within a reasonable number of iterations. A rough estimation of the necessary steps for convergence is given in [8], using the CFL time step restriction for the explicit scheme used by BBS. The number of steps to reach a stationary state is estimated as large as $N_{pixels}^{1/2}$, which results in several thousands of iterations for typical test cases. Our implementation of the original method displays convergence properties of this order of magnitude.

Our method does not display these shortcomings related to the practical implementation: there are not constants to be tuned, and no advanced knowledge of numerical methods for

Navier–Stokes equations is required. Since we use a minimization technique, we can choose a locally minimizing time step to ensure that the energy is minimized. In addition, by preconditioning the Euler–Lagrange equations, we remove the need to apply anisotropic diffusion. In contrast with the method of BBS, the results for inpainting with and without anisotropic diffusion were nearly identical for the test cases we tried.

We first evaluate convergence properties of our method, compared to the method of BBS, for the test case displayed in Figure 2. Two minimization schemes were used, based upon the Euler–Lagrange equations and the Sobolev gradient using just one application of the preconditioner (also referred to as the H^3 gradient), respectively. For the purpose of demonstrating the effect of preconditioning on image inpainting, we found it sufficient and reasonable to apply the preconditioner $(I - \Delta)^{-1}$ just once. Thus in the later images, in order to save CPU time, we used just one application of preconditioning.

We plot in Figure 1 the reduction in $\|\nabla^\perp u \cdot \nabla \Delta u\|^2$ using the method of BBS and our energy minimization method. We can observe that our method is quantitatively closer to the solution of the PDE. With our minimization scheme we were able to achieve a decrease of two or three orders of magnitude for most cases. As our method is a gradient descent method, we expect a linear rate of convergence. However, we do note that by using the Sobolev gradient in place of the Euler–Lagrange equations, the convergence to zero is slightly accelerated.

In Figure 1 we also plot the “error” $\varepsilon = \max |u_{n+1} - u_n|$, where u_n represents the current image. This quantity is used as the stopping criterion for the iterative process since it seems to be the best measure for convergence in image processing, as it is generally a valid measure of convergence in numerical analysis. We should note in passing that, compared to fluid dynamics applications where $\varepsilon < 10^{-8}$ is required for convergence, for image processing applications no apparent change in the image is observed starting with $\varepsilon < 10^{-4}$.

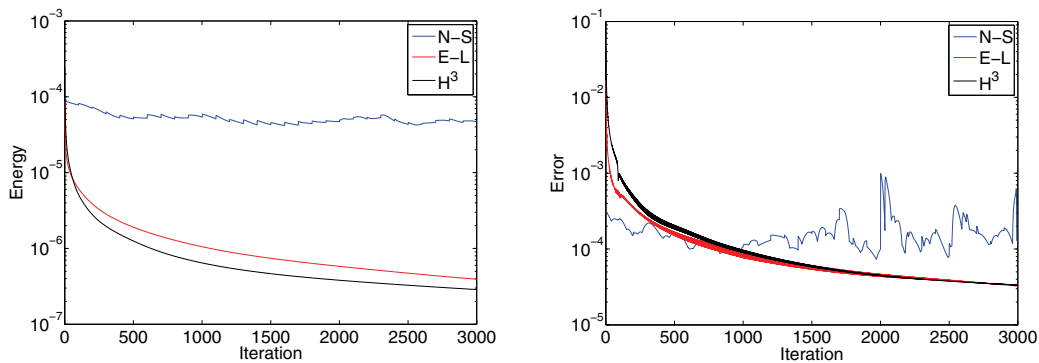


Figure 1. Left: comparison of the norm of the residual (energy) for the different methods. Right: the error as defined by $\max |u_{n+1} - u_n|$ for the different methods. N–S denotes the Navier–Stokes method following the implementation of BBS. E–L denotes the iterative method using the Euler–Lagrange equations. H^1 denotes the Sobolev gradient.

An important question related to convergence is the condition number of the algorithm. Although the condition number of the least-squares problem when one considers minimization with the Lagrange equations is relatively high, we demonstrate that this number is significantly reduced when we precondition the problem and take the gradient in a higher-order Sobolev

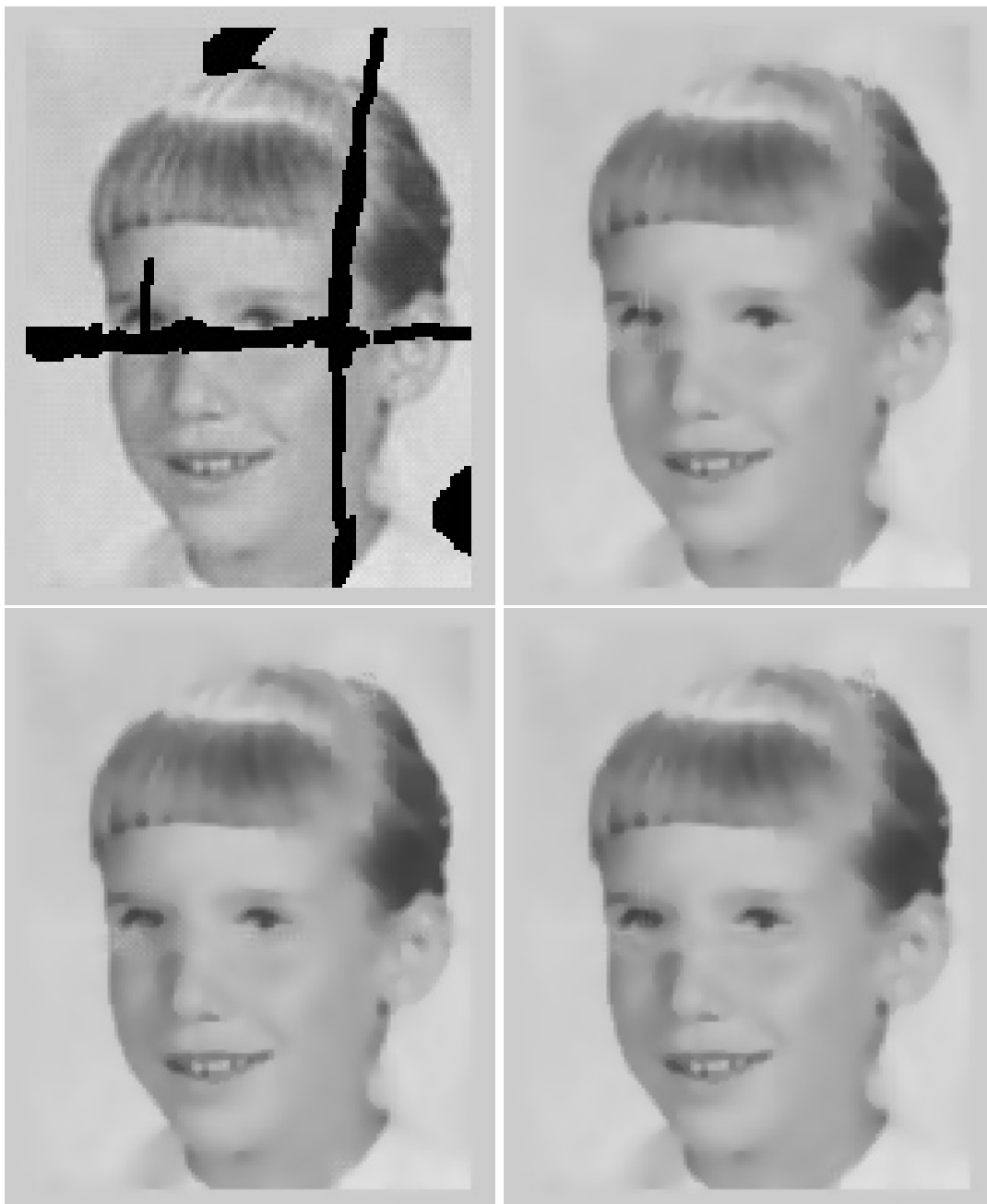


Figure 2. *The resulting images for the simulations. Top left: the defective image. Top right: Navier–Stokes inpainting. Bottom left: energy minimization with the Euler–Lagrange equations. Bottom right: energy minimization with the H^1 gradient. Based on original image from Amy Miller, The Photo Medic.*

space. Since these results concern more general numerical analysis considerations, we report them in the appendix.

The images in Figure 2 were run until an error tolerance of 10^{-4} was achieved. Using

the method of BBS, this took 9282 iterations in a CPU time of 67.28 seconds. Using energy minimization with the Euler–Lagrange equations, this took 1014 steps in a CPU time of 29 seconds. Using one application of the preconditioner, convergence was reached in 742 iterations in a CPU time of 32 seconds. Using three applications, convergence was reached in 942 steps in about 40 seconds. We remark that by preconditioning we were able to reach the error tolerance using fewer iterations. The CPU time was comparable for the case using the Euler–Lagrange equations and the Sobolev gradient, as solving a linear system at each time step adds to the processing time. However, one could take further measures to decrease this additional time by implementing a different algorithm for solving the linear system.

Qualitatively, the effect of preconditioning is apparent in the bottom two images in Figure 2. While the image resulting from the Euler–Lagrange equations has some noise where the inpainting was performed, the image resulting from the Sobolev gradient is smooth in the inpainting region. Also note that the edge between the face and the background is correctly placed. In this sense, we remove the need to apply anisotropic diffusion.

In this image the result we present for the BBS method is qualitatively worse than the result presented in the original paper of Bertalmío et al. [7]. We attribute this to the fluctuating error as presented in Figure 1; the result for this problem initially improved, but the subsequent images did not correctly inpaint the eye. This behavior is certainly related to the careful choice of the parameters in the original implementation (time step, when and how to apply the anisotropic diffusion). Since the error analysis is not available for the method of BBS, we did not attempt to find a deterministic choice for these parameters.

Since the full image of the girls, which first appeared in [7], is currently used to test inpainting methods, we present in Figure 3 our results for this test case. The inpainting region is correctly filled, with a better image quality obtained when the Sobolev gradient method is used.

6.2. Qualitative comparison with other methods. In an effort to compare our results to other methods used in image inpainting, here we use Figure 4 (see [8]). We performed our inpainting algorithm on just the grayscale image as the issues of edge recovery and smoothness of the interpolation can be determined from this image.

In comparison with the method of Bornemann and März [8], the construction of the iris is comparable. In their result a better quality image was achieved as the texture of the eyebrow was interpolated. Also, the CPU for this example was about 60 seconds. In [8] a reported CPU time of 0.4 seconds was reported. It goes without saying that our iterative method could not compete in terms of efficiency with a noniterative fast-marching method, such as that proposed in [8]. Nevertheless, the method of Bornemann and März needs a careful tuning of four parameters, which is not the case for our method (see also [6]).

6.3. Image interpolation. In this section, we apply our algorithm to perform image interpolation. We start with the two images given in Figure 5 and interpolate the bird to 160 by 152 pixels and Lena to 200 by 256 pixels using nearest neighbor interpolation in Figure 5. The interpolation is carried out as follows. The image is first expanded to the dimensions of the larger image, where the newly added pixels designate the inpainting region. As with inpainting, we first solve Laplace’s equation and then apply the inpainting algorithm to further sharpen the image.

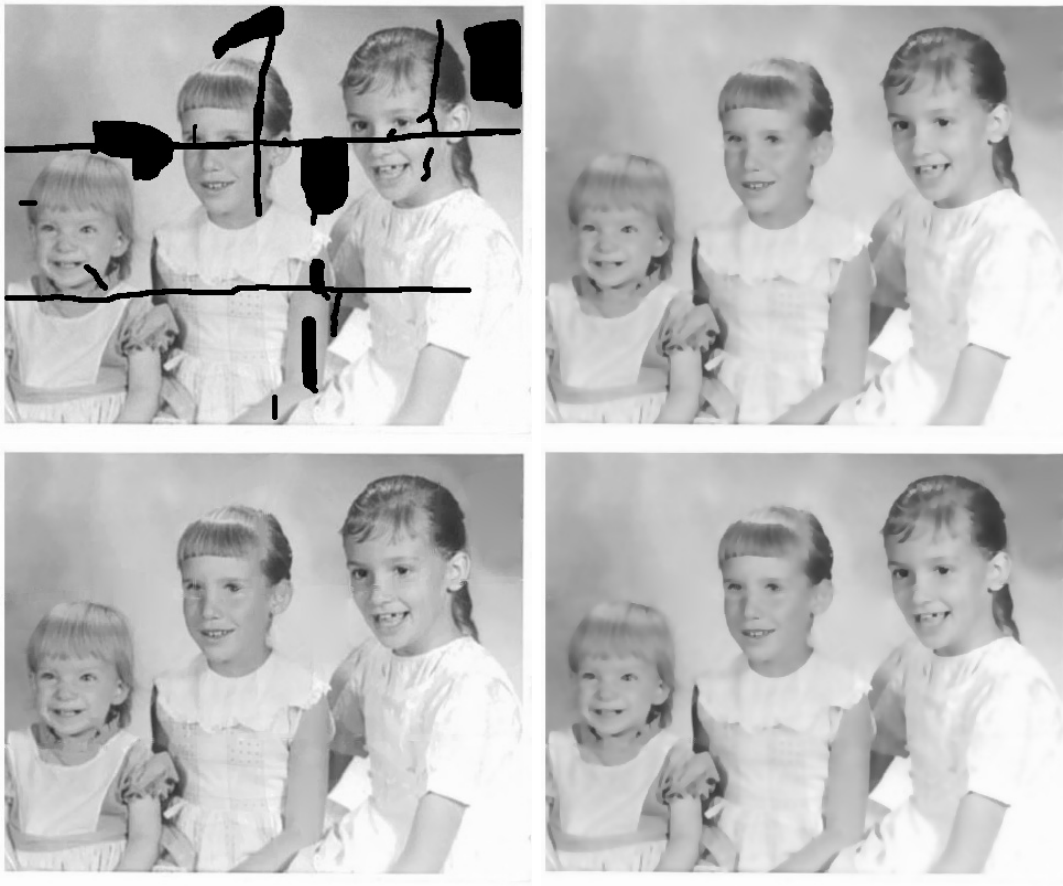


Figure 3. *Inpainting of the girls. Compare with the original papers [7] and [8]. Top left: original image. Top right: the solution of Laplace equation. Bottom left: the solution using Navier–Stokes inpainting. Bottom right: energy minimization with the H^1 gradient. Original photo from Amy Miller, The Photo Medic.*



Figure 4. *Inpainting of the eye. Left: the solution to Laplace's equation. Center: the solution using Navier–Stokes inpainting. Right: energy minimization with the H^1 gradient. For comparison with previous works, see [8] and [25].*

For these two images the effect of using anisotropic diffusion in the method of BBS is evident. The smoothing effect of anisotropic diffusion does not allow one to recover the detail of these two images. In the energy minimization method, since we avoid using anisotropic



Figure 5. Image interpolation: original image enlarged using nearest neighbor interpolation. Left: bird original size 80 by 76 pixels, Right: Lena original image 50 by 64 pixels.

diffusion, we can recover sharp edges as well as some of the finer details. For example, in the image of the bird, notice that in the energy minimization method using both the Euler–Lagrange equation and the Sobolev gradient, the eye of the bird was actually recovered. The effect of preconditioning here is evident in that it minimizes some of the artifacts associated with image interpolation. For example, comparing the beak of the bird in the image resulting from cubic interpolation and minimization with the Lagrange equation versus minimization with the Sobolev gradient, we see that the latter results in sharper edges. Similar effects are evident when we examine the interpolation of Lena. In particular, the smoothing and sharper edges are most visible in the upper part of the hat of Lena.

We remark here that in practice the assumption that u be C^1 continuous on the entire domain $\bar{\Omega}$ is too strong. In image inpainting an edge represents a discontinuity; thus in theory this important case would be excluded from our analysis. The idea of using a gradient in a higher-order Sobolev space is to increase the smoothness of the interpolation, and this is supported by the theory as when one computes a gradient in a higher-order Sobolev space, we see that the minimizer should also satisfy further regularity conditions. In several of our cases, using a Sobolev gradient reduced artifacts associated with image inpainting and image interpolation while preserving edges (see Figures 6, 7, and 2 and the associated discussions). However, some care should be taken in practice as oversmoothing would result in blurred reconstruction of edges. This artifact appeared in some of our results (see, for example, 4). In principle, the analysis of this problem could be studied and carried out in a lower-order Sobolev space or in the space of functions of bounded variation, which is the traditional space used for



Figure 6. Resulting images for image interpolation. Interpolation to twice as many pixels. Top left: result from the method of Bertalmío et al. [4]. Top right: result from bicubic interpolation. Bottom left: result from minimization using the Euler equations. Bottom right: result from minimization using the H^1 gradient. Notice that artifacts are not present when using the H^1 gradient (see beak of bird, for example).

an analytical study of problems in image processing. We expect that the estimates required for such an analysis would be of a similar nature to the estimates required for existence and uniqueness results of the Navier–Stokes equations and hence would be beyond the scope of this paper.

7. Summary and final discussion. In this work we derived a method for solving the digital image inpainting model that is based on the model of the Navier–Stokes equations for an incompressible flow. We used a variational formulation and a gradient flow to converge to the stationary solution. The flow is based on a gradient coming from a Sobolev norm, instead of the generally used L^2 norm, which in practice is associated with preconditioning the Euler–Lagrange equations. We found that through the use of the preconditioner, we can favorably alter properties of the interpolation by reducing noise and offering a more accurate interpolation of the edges.



Figure 7. Resulting images for image interpolation. Interpolation to four times as many pixels. Top left: result from the method of Bertalmio et al. [4]. Top right: result from bicubic interpolation. Bottom left: result from minimization using the Euler equations. Bottom right: result from minimization using the H^1 gradient.

Compared to the original Navier–Stokes model proposed by Bertalmío, Bertozzi, and Sapiro [5], our algorithm displays better convergence properties, with similar qualitative image inpainting results. From the practical point of view of computer implementation, we have eliminated technical aspects related to the use of numerical methods for Navier–Stokes equations (slope limiters, dynamic relaxation for Poisson equation, anisotropic diffusion steps, etc.). We also mention that the new algorithm does not require us to tune constants as in noniterative methods [8]. Using a straightforward finite difference implementation, we demonstrated, through various examples for image inpainting and image interpolation, that the algorithm is effective for such problems.

Finally we would like to emphasize that the mathematical theoretical framework that we developed in this paper is of more general interest and could be applied to other variational formulations used in image inpainting, such as the one presented in [17]. Such new applications of the Sobolev gradient appear as a natural extension of the present work.

Appendix. Condition number. We give two precise definitions for the condition number and show that by Sobolev preconditioning, the relative condition number is improved, as shown in Figure 8.

Following the definition given in [27], the following definitions for the condition number of a function f from an arbitrary normed space X to an arbitrary normed space Y are used.

Definition A.1. *The relative condition number of a differentiable function $f : X \rightarrow Y$ is defined as*

$$\frac{\|f'(x)\|_{L(X,Y)}\|x\|_X}{\|f(x)\|_Y}.$$

Definition A.2. *The absolute condition number of a function $f : X \rightarrow Y$ is given by*

$$\kappa = \lim_{\delta \rightarrow 0} \sup_{\|h\|_X \leq \delta} \frac{\|f(x+h) - f(x)\|}{\|h\|_X}.$$

$\|f'(x)\|_{L(X,Y)}$ is the norm of $f'(x)$ when viewed as a bounded linear operator from X to Y for a differentiable function f . If $X = \mathbb{R}^n$ is equipped with the Euclidean metric and $Y = \mathbb{R}$, then $f'(x)$ is just the Jacobian of f at x .

Note that if $Y = \mathbb{R}$, then $f'(x)h = \langle h, \nabla_X f(x) \rangle_X$ for each $x \in X$, following the definition of the Sobolev gradient. Thus $\|f'(x)\|_{L(X,\mathbb{R})} = \sup\{f'(x)h : h \in X, \|h\|_X = 1\} = \|\nabla_X f(x)\|_X$.

In the finite dimensional setting, H^k is the space of all n -dimensional vectors equipped with the metric $\langle u, v \rangle_{H^k} = \langle u, (I - \Delta)^k v \rangle_{\mathbb{R}^n}$. Here Δ is a discretization of the Laplacian. Let $f : H^k \rightarrow \mathbb{R}$; then using the definition of the gradient,

$$f'(x)h = \langle h, \nabla f(x) \rangle_{\mathbb{R}^n} = \langle h, (I - \Delta)^k \nabla_{H^k} f(x) \rangle_{\mathbb{R}^n} \quad \forall h \in \mathbb{R}^n.$$

Thus

$$(A.1) \quad \nabla_{H^k} f(x) = (I - \Delta)^{-k} \nabla f(x),$$

and

$$\|\nabla_{H^k} f(x)\|_{H^k} = \langle (I - \Delta)^k (I - \Delta)^{-k} \nabla f(x), (I - \Delta)^{-k} \nabla f(x) \rangle = \langle \nabla f(x), (I - \Delta)^{-k} \nabla f(x) \rangle.$$

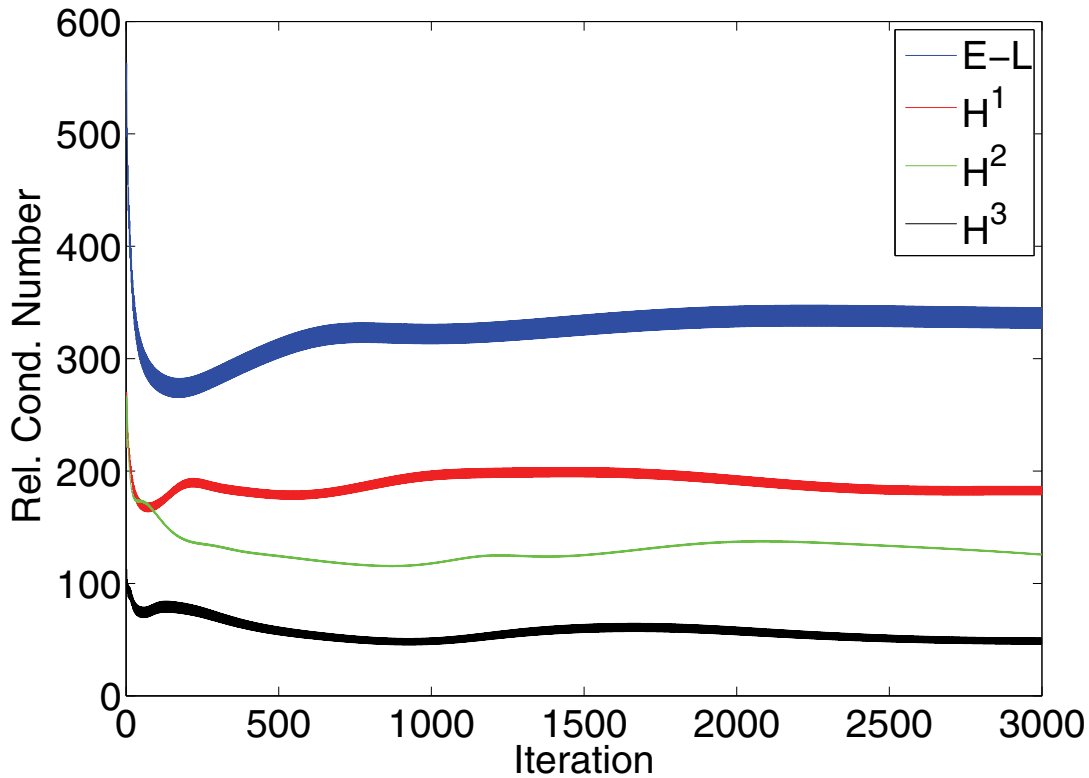


Figure 8. Plot of the relative condition number versus iteration corresponding to the inpainting of Figure 2. E–L corresponds to the minimization of the energy with the Euler–Lagrange equation (no preconditioning), and H^1 , H^2 , and H^3 correspond to the minimization using the H^1 , H^2 , or H^3 gradients.

We make two remarks. First, we see that by equipping the domain of the problem with the H^k norm in place of the Euclidean norm, we precondition the problem so that the absolute condition number of the problem as defined above is reduced by a factor of $|(I - \Delta)^{-1}|^k$.

This effect can be easily illustrated if the Fourier representation of the gradient is used. If the ordinary gradient is sampled on a rectangular, regular grid as

$$(A.2) \quad \nabla f(x_j) = \sum_p \hat{g} e^{i\lambda_p x_j},$$

then, following (A.1), the H^k gradient will have a similar representation (same basis functions) with modified Fourier coefficients:

$$\nabla_{H^k} f(x_j) = \sum_p \hat{G} e^{i\lambda_p x_j}, \quad \hat{G} = \frac{\hat{g}}{(1 + \lambda_p^2)^k}.$$

Since $k \geq 1$, we infer that $\|\nabla_{H^k} f(x)\|^2 = \sum \|\hat{G}\|^2$ is reduced and therefore the condition number is improved. The practical consequence is that diminishing the corresponding wave numbers in the Fourier decomposition results in attenuating the modes which oscillate the

most. This could explain the smoothing effect of Sobolev gradient methods (see also [15]). Second, using the above reasoning, if f is real valued and $X = H^k$, then $\|f'(x)\|_{L(X,Y)} = \|\nabla_X f(x)\|_X$, and hence the relative condition number is given by

$$\frac{\langle \nabla f(x), (I - \Delta)^{-k} \nabla f(x) \rangle \|x\|_{H^k}}{|f(x)|}.$$

In the examples we present, we normalize all images so that the maximum pixel value is one and thus use the relative condition number as a measure of how well conditioned the problem is.

In Figure 8, we give a plot of the relative condition number for the case corresponding to Figure 2. We compare the relative condition number of the problem with and without preconditioning. The relative condition number is lowest for H^3 , achieving a reduction of a factor of 4 compared with the Lagrange equations.

Another possibility for dealing with the condition number issues is to reformulate this problem as a first-order system. This method is currently under investigation in [22] and possibly could be applied to the inpainting problem.

Acknowledgments. We thank Dr. Santiago Betelu for suggesting this problem. We would also like to acknowledge J. W. Neuberger, S. Masnou, and F. Hecht for stimulating discussions and useful suggestions.

REFERENCES

- [1] R. A. ADAMS AND J. F. FOURNIER, *Sobolev Spaces*, Elsevier Academic Press, Amsterdam, 2003.
- [2] P. ARIAS, V. CASELLES, AND G. SAPIRO, *A variational framework for non-local image inpainting*, in Energy Minimization Methods in Computer Vision and Pattern Recognition, Lecture Notes in Comput. Sci. 5681, Springer-Verlag, 2009, pp. 345–358.
- [3] G. AUBERT AND P. KORNPORST, *Mathematical Problems in Image Processing. Partial Differential Equations and the Calculus of Variations*, Appl. Math. Sci. 147, Springer-Verlag, New York, 2002.
- [4] C. BALLESTER, M. BERTALMIÓ, V. CASELLES, G. SAPIRO, AND J. VERDERA, *Filling-in by joint interpolation of vector fields and gray levels*, IEEE Trans. Image Process., 10 (2001), pp. 1200–1211.
- [5] M. BERTALMIÓ, A. L. BERTOZZI, AND G. SAPIRO, *Navier-Stokes, fluid dynamics, and image and video inpainting*, in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2001, pp. 355–362.
- [6] M. BERTALMIÓ, V. CASELLES, S. MASNOU, AND G. SAPIRO, *Inpainting*, in Encyclopedia of Computer Vision, Springer, to appear.
- [7] M. BERTALMIÓ, G. SAPIRO, V. CASELLES, AND C. BALLESTER, *Image inpainting*, in Proceedings of the 27th Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '00), ACM Press/Addison-Wesley, New York, 2000, pp. 417–424.
- [8] F. BORNEMANN AND T. MÄRZ, *Fast image inpainting based on coherence transport*, J. Math. Imaging Vision, 28 (2007), pp. 259–278.
- [9] J. CALDER, A. MANSOURI, AND A. YEZZI, *Image sharpening via Sobolev gradient flows*, SIAM J. Imaging Sci., 3 (2010), pp. 981–1014.
- [10] J. CALDER, A. MANSOURI, AND A. YEZZI, *New possibilities in image diffusion and sharpening via high-order Sobolev gradient flows*, J. Math. Imaging Vision, 40 (2011), pp. 248–258.
- [11] T. F. CHAN, S. H. KANG, AND J. SHEN, *Euler's elastica and curvature-based inpainting*, SIAM J. Appl. Math., 63 (2002), pp. 564–592.
- [12] T. F. CHAN AND J. SHEN, *Image Processing and Analysis: Variational, PDE, Wavelet, and Stochastic Methods*, SIAM, Philadelphia, 2005.

- [13] I. DANAILA AND P. KAZEMI, *A new Sobolev gradient method for direct minimization of the Gross–Pitaevskii energy with rotation*, SIAM J. Sci. Comput., 32 (2010), pp. 2447–2467.
- [14] A. CRIMINISI, P. PÉREZ, K. TOYAMA, *Region filling and object removal by exemplar-based image inpainting*, IEEE Trans. Image Process., 13 (2004), pp. 1200–1213.
- [15] J. J. GARCÍA-RIPOLL AND V. M. PÉREZ-GARCÍA, *Optimizing Schrödinger functionals using Sobolev gradients: Applications to quantum mechanics and nonlinear optics*, SIAM J. Sci. Comput., 23 (2001), pp. 1316–1334.
- [16] P. KAZEMI AND J. W. NEUBERGER, *Potential theory and applications in a constructive method for finding critical points of Ginzburg–Landau type equations*, Nonlinear Anal., 69 (2008), pp. 925–930.
- [17] S. MASNOU AND J. M. MOREL, *Level lines based disocclusion*, in Proceedings of the 5th IEEE International Conference on Image Processing, Chicago, IL, 1998, pp. 259–263.
- [18] J. W. NEUBERGER, *Sobolev Gradients and Differential Equations*, Springer-Verlag, Berlin, 2010.
- [19] M. NITZBERG, D. MUMFORD, AND T. SHIOTA, *Filtering, Segmentation, and Depth*, Springer-Verlag, Berlin, 1993.
- [20] P. PERONA AND J. MALIK, *Scale-space and edge detection using anisotropic diffusion*, IEEE Trans. Pattern Anal. Mach. Intell., 12 (1990), pp. 629–639.
- [21] R. J. RENKA, *Image segmentation with a Sobolev gradient method*, Nonlinear Anal., 71 (2009), pp. 774–780.
- [22] R. J. RENKA, *A Sobolev gradient method for treating the steady-state incompressible Navier–Stokes equation*, submitted.
- [23] W. B. RICHARDSON, *Sobolev gradient preconditioning for image-processing PDEs*, Comm. Numer. Methods Engrg., 24 (2008), pp. 493–504.
- [24] G. SAPIRO, *Geometric Partial Differential Equations and Image Analysis*, Cambridge University Press, Cambridge, UK, 2001.
- [25] A. TELEA, *An image inpainting technique based on the fast marching method*, J. Graphics Tools, 9 (2004), pp. 23–34.
- [26] R. TEMAM, *Navier–Stokes Equations and Nonlinear Functional Analysis*, CBMS-NSF Regional Conf. Ser. Appl. Math. 66, 2nd ed., SIAM, Philadelphia, 1995.
- [27] L. N. TREFETHEN AND D. BAU, III, *Numerical Linear Algebra*, SIAM, Philadelphia, 1997.
- [28] D. TSCHUMPERLÉ, *Fast anisotropic smoothing of multi-valued images using curvature-preserving PDE’s*, Int. J. Comput. Vision, 68 (2006), pp. 65–82.
- [29] D. TSCHUMPERLÉ AND R. DERICHE, *Vector-valued image regularization with PDEs: A common framework for different applications*, IEEE Trans. Pattern Anal. Mach. Intell., 27 (2005), pp. 506–517.
- [30] J. WEICKERT, *Anisotropic Diffusion in Image Processing*, Teubner, Stuttgart, Germany, 1998.