

A Fractional Inpainting Model Based on the Vector-Valued Cahn–Hilliard Equation*

Jessica Bosch[†] and Martin Stoll[‡]

Abstract. The Cahn–Hilliard equation provides a simple and fast tool for binary image inpainting. By now, two generalizations to gray value images exist: bitwise binary inpainting and TV- H^{-1} inpainting. This paper outlines a model based on the vector-valued Cahn–Hilliard equation. Additionally, we generalize our approach to a fractional-in-space version. Fourier spectral methods provide efficient solvers since they yield a fully diagonal scheme. Furthermore, their application to three spatial dimensions is straightforward. Numerical examples show the superiority of the proposed fractional Cahn–Hilliard inpainting approach over its nonfractional version. It improves the peak signal-to-noise ratio and structural similarity index. Likewise, the experiments confirm that the proposed model competes with previous inpainting methods, such as the total variation inpainting approach and its fourth-order variant.

Key words. image inpainting, gray value images, Cahn–Hilliard equation, vector-valued problems, fractional calculus, fractional Laplacian

AMS subject classifications. 65N35, 93C20, 35R35, 35K55, 82C26, 35R11, 65T50

DOI. 10.1137/15M101405X

1. Introduction. Three-dimensional visualizations of medical images help professionals make more accurate diagnoses. Parts of the human body are given in the form of a sequence of slices. Often, the distance between these slices is significantly larger than the image pixel size. Hence, it is necessary to interpolate additional slices in order to obtain an accurate three-dimensional description. This can be achieved by the image inpainting process [57, 11, 38].

Inpainting is the art of modifying parts of an image such that the resulting changes are not easily detectable by an ordinary observer. Applications include the restoration of damaged paintings and photographs [3], the replacement of selected objects, and the reduction of artifacts in medical images [39]. Due to the large number of applications, much effort has gone into the development of digital inpainting techniques—starting with classical integer-order models through to generalized fractional-order approaches. This section presents a brief survey of such methods.

1.1. Classical inpainting models. Bertalmío et al. [6] introduced the image inpainting technique into digital image processing. Their third-order nonlinear PDE propagates the

*Received by the editors March 26, 2015; accepted for publication (in revised form) August 12, 2015; published electronically October 20, 2015. This work was supported by a research grant of the “International Max Planck Research School (IMPRS) for Advanced Methods in Process and System Engineering (Magdeburg).”

<http://www.siam.org/journals/siims/8-4/M101405.html>

[†]Corresponding author. Numerical Linear Algebra for Dynamical Systems, Max Planck Institute for Dynamics of Complex Technical Systems, 39106 Magdeburg, Germany (bosch@mpi-magdeburg.mpg.de).

[‡]Numerical Linear Algebra for Dynamical Systems, Max Planck Institute for Dynamics of Complex Technical Systems, 39106 Magdeburg, Germany (stollm@mpi-magdeburg.mpg.de).

image information along the lines of equal gray values into the areas to be modified. Consecutively, a number of variational- and PDE-based approaches have been considered, including the total variation (TV) inpainting model [21, 19], inpainting based on curvature-driven diffusions [20], Euler’s elastica inpainting model [18], the Mumford–Shah inpainting model [62, 21], and the Mumford–Shah–Euler inpainting model [32]. We refer the reader to [22] for an overview of existing inpainting methods. In summary, Schönlieb and Bertozzi [61] pointed out that higher order inpainting methods have certain advantages over second-order ones. These include, for example, the preservation of curvatures or the connection of contours across very large distances.

1.2. Fractional inpainting models. In recent years, fractional-order PDEs have been studied in image processing [67, 4]. Fractional differential equations replace a standard differential operator by a corresponding fractional differential operator. Generalizations to integrals and derivatives of arbitrary order have a long history in mathematics [55]. They are used to describe many real-world phenomena, e.g., the anomalous diffusion process [53], dispersion in heterogeneous aquifers [1], cardiac electrical propagation [15], filtration of solutes in porous soils [5], and receptor-mediated transport of morphogens in developing tissues [45]. In general, fractional differential equations characterize nonlocal and spatial heterogeneous properties where classical models provide rather inadequate results. For inpainting problems, they improve the image quality and peak signal-to-noise ratio (PSNR) [4, 69, 46, 68]. The new fractional schemes can be treated as regularizations of the classical inpainting models and overcome the well-known effect of oversmoothing. Similar effects have been studied in regularization methods for ill-posed problems [44, 41], which are applied among other things in denoising.

1.3. Basic model. Bertozzi, Esedoğlu, and Gillette [8, 7] introduced the fourth-order Cahn–Hilliard inpainting approach for binary images. They benefit from the use of the fast Fourier transform (FFT) to achieve fast inpainting. Compared to previous classical models, their approach is based on a simplified PDE and reduces the computational time.

To our knowledge, two generalizations to gray value images have been studied: Schönlieb and Bertozzi’s [61] bitwise binary inpainting and Burger, He, and Schönlieb’s [16] TV- H^{-1} inpainting. First, the bitwise binary inpainting method splits the image bitwise into channels. The Cahn–Hilliard inpainting approach applies then to each binary channel separately. Second, the TV- H^{-1} inpainting model adds subgradients of the TV functional. This fourth-order variant of the TV inpainting model is shown to be superior to the original second-order TV model.

In this paper, we present a vector-valued Cahn–Hilliard inpainting model for gray value images. Not only does this model consist of a number of scalar Cahn–Hilliard equations, but it also connects all of them via a coupling term. In fact, this term arises during the derivation of the system of PDEs via a projected gradient flow. This projection ensures that the solution stays an element of the Gibbs simplex. We refer the reader to section 3 for more details regarding the derivation. Further, section 6.5 demonstrates numerically the fulfillment of the Gibbs simplex condition. The second target of the present paper is the generalization of our proposed model to a fractional-in-space version, called fractional Cahn–Hilliard inpainting. Numerical examples emphasize the superiority of the proposed fractional

Cahn–Hilliard inpainting approach over its nonfractional version. This generalized approach shows a sharpness effect and improves the image quality. In particular, we observe an increase of the PSNR and the structural similarity (SSIM) values. Likewise, comparisons to previous models including TV and TV- H^{-1} inpainting as well as bitwise binary inpainting support our model.

The paper is organized as follows. Section 2 briefly summarizes Bertozzi, Esedoğlu, and Gillette’s [8, 7] binary Cahn–Hilliard inpainting model. It motivates the use of the Cahn–Hilliard equation by describing its main features. A vector-valued formulation based on the multicomponent Cahn–Hilliard equation is presented in section 3. In particular, we highlight the differences to the scalar version. A common feature is that both models do not follow a variational principle. However, Schönlieb and Bertozzi [61] pointed out that the idea of convexity splitting [31, 34] can be applied to evolution equations that do not follow a variational principle. Hence, we apply this technique to the vector-valued formulation in section 4. As a result, we get an unconditionally gradient stable time-discrete scheme. Before deriving the fully discrete system, we generalize it to a fractional-in-space version. In section 5, we exploit the spectral decomposition of the occurring (fractional) Laplace operator as proposed by Bueno-Orovio, Kay, and Burrage [14]. Hence, we can take advantage of the FFT similar to the binary inpainting approach [8]. Simulation results are shown in section 6. The test images include, besides simple sharp stripe images, a fingerprint and a landscape. Moreover, we show the performance of Cahn–Hilliard inpainting for the metal artifact reduction in computed tomography (CT) images. Finally, we perform three-dimensional visualizations of magnetic resonance imaging (MRI) images that comprise a scan of a human cranium. Section 7 summarizes our findings. At last, we study the numerical analysis of the nonfractional Cahn–Hilliard inpainting approach. These appendices are provided as supplementary material with this paper (see M101405_01.pdf [local/web 98.8KB]). We derive the convexity conditions for the application of the convexity splitting method in Appendix A. Finally, in Appendix B we prove the consistency, unconditional stability, and convergence of the time-discrete scheme.

2. Binary Cahn–Hilliard inpainting. We start by reflecting on Bertozzi, Esedoğlu, and Gillette’s [8, 7] black-and-white Cahn–Hilliard inpainting model. Let f be the given binary image which is defined on the image domain $\Omega \subset \mathbb{R}^d$ ($d = 2, 3$). Let $T > 0$ be a fixed time. The parts of f that are going to be modified are denoted by the inpainting domain $D \subset \Omega$. The target is to reconstruct the image in this region D in an undetectable way. The inpainted version $u: \Omega \times (0, T) \rightarrow \mathbb{R}$ of f is constructed by following the evolution of

$$(2.1) \quad \partial_t u = -\Delta \left(\varepsilon \Delta u - \frac{1}{\varepsilon} \psi'(u) \right) + \omega(f - u),$$

$$(2.2) \quad \nabla u \cdot \mathbf{n} = \nabla(\Delta u) \cdot \mathbf{n} = 0 \quad \text{on } \partial\Omega,$$

where

$$(2.3) \quad \omega = \omega(\mathbf{x}) = \begin{cases} 0 & \text{if } \mathbf{x} \in D, \\ \omega_0 & \text{if } \mathbf{x} \in \Omega \setminus D \end{cases}$$

is the fidelity parameter. Bertozzi, Esedoğlu, and Gillette [8, 7] named this model the modified Cahn–Hilliard equation. The choice $\omega \equiv 0$ leads to the original Cahn–Hilliard equation, which

is a well-known model for coarsening and phase separation processes [40, 17]. Before we discuss the characteristics of the inpainting model (2.1)–(2.2) we briefly present its derivation. It basically arises from a superposition of two gradient flows: an H^{-1} -gradient flow for the Cahn–Hilliard part and an L^2 -gradient flow for the fidelity term.¹

2.1. Cahn–Hilliard equation. The Cahn–Hilliard equation models the motion of interfaces between two phases where the total concentration is assumed to be conserved [40, 17]. The underlying energy functional is based on the Ginzburg–Landau energy $\mathcal{E}_1: H^1(\Omega) \rightarrow \mathbb{R}$:

$$(2.4) \quad \mathcal{E}_1(u) = \int_{\Omega} \left\{ \frac{\varepsilon}{2} |\nabla u|^2 + \frac{1}{\varepsilon} \psi(u) \right\} d\mathbf{x}.$$

Now, the process evolves such that \mathcal{E}_1 decreases in time. Before we describe the way of minimizing the energy \mathcal{E}_1 , we explain the underlying system and the terms in \mathcal{E}_1 . The phase variable $u \in H^1(\Omega)$ describes the concentration of two phases, e.g., two components of an alloy. Let us denote these two phases by A and B . If $u(\mathbf{x}, t) = 0$, then only phase A is present at point \mathbf{x} at time t . The case $u(\mathbf{x}, t) = 1$ means only phase B exists at (\mathbf{x}, t) . Values of u between zero and one represent mixed regions. These areas form the interface, which acts as a diffuse phase transition. We can control its thickness via the model parameter $\varepsilon > 0$. Usually, the aim is to keep it as small as possible. The evolution of the interface is driven by the interfacial energy $\frac{\varepsilon}{2} |\nabla u|^2$ (see (2.4)), whose minimization penalizes interfaces with high curvatures.

The potential function ψ in (2.4) gives rise to phase separation. It has two minima at zero and one. Since ψ is a function of u it becomes minimal at those points where either only phase A or only phase B is present. Hence, its minimization penalizes mixed regions. Well-known potential functions are logarithmic potentials [17, 27, 26], smooth double-well potentials [29, 8, 50, 25], and double-obstacle potentials [56, 13]. Bertozzi, Esedoğlu, and Gillette [8, 7] used the smooth double-well potential

$$(2.5) \quad \psi(u) = u^2(u - 1)^2.$$

In this work, we concentrate on this type of potential as well.²

Now, we come back to the minimization of the energy functional \mathcal{E}_1 in time. A standard way of doing this is the gradient flow approach. This lets u evolve in the opposite direction to the gradient of \mathcal{E}_1 at u with respect to an inner product corresponding to a vector space Z . The gradient flow of \mathcal{E}_1 with respect to the inner product $(\cdot, \cdot)_Z$ is given as

$$\partial_t u(t) = -\text{grad}_Z \mathcal{E}_1(u(t)).$$

The use of the mass conserving H^{-1} -inner product

$$(v_1, v_2)_{H^{-1}(\Omega)} := \int_{\Omega} \nabla(-\Delta)^{-1} v_1 \cdot \nabla(-\Delta)^{-1} v_2 \, d\mathbf{x},$$

¹Hence, the modified Cahn–Hilliard equation as a whole is not strictly a gradient flow.

²Note that we have recently developed fast iterative solvers for the use of a nonsmooth double-obstacle potential [13]. This was done for the black-and-white Cahn–Hilliard inpainting model. The generalization of the developed solver to the proposed vector-valued inpainting model is also recently done by the same authors. A discussion of this topic would go beyond the scope of the present paper. We refer the reader to [12], which discusses fast iterative solvers for smooth and nonsmooth vector-valued Cahn–Hilliard equations.

where

$$(2.6) \quad H^{-1}(\Omega) = \{u^* \in H^1(\Omega)^* \mid \langle u^*, 1 \rangle_{H^1(\Omega)^*, H^1(\Omega)} = 0\},$$

leads us to the original Cahn–Hilliard equation, i.e., (2.1)–(2.2) with $\omega \equiv 0$. Here, $H^1(\Omega)^*$ is the dual space of $H^1(\Omega)$ with the dual pairing $\langle \cdot, \cdot \rangle_{H^1(\Omega)^*, H^1(\Omega)}$. The operator Δ^{-1} denotes the inverse of Δ with Neumann boundary conditions. Finally, the Cahn–Hilliard equation is imposed with the natural zero Neumann boundary condition as well as the mass-conserving boundary condition (see (2.2)). Here, \mathbf{n} is the unit normal vector to $\partial\Omega$ pointing outward from Ω . We refer the reader to Blank, Butz, and Garcke [10] and the references therein for more details about the derivation.

As mentioned in the previous section, Bertozzi, Esedoğlu, and Gillette’s [8, 7] black-and-white Cahn–Hilliard inpainting model arises from a superposition of two gradient flows: an H^{-1} -gradient flow for the Cahn–Hilliard part and an L^2 -gradient flow for the fidelity term. We have finished the discussion of the first part. Now, we want to briefly look into the second part.

2.2. Modification of the Cahn–Hilliard equation. In the previous section, we have explained the underlying system in general terms of phases. Remember that we have called u the phase variable which describes the concentration of two phases A and B . Now, we want to explain the underlying system in terms of image processing. In this setting, the two phases A and B represent the colors black and white. Their interface consists of gray values and forms a smooth transition. Basically, we can imagine u as a black-and-white image that evolves in time. The initial image is set to be $u(\mathbf{x}, 0) = f$. Note that f was a given black-and-white image which contains some damaged regions. These regions are collected in the set $D \subset \Omega$. The task is to reconstruct the image f in this region D in an undetectable way. The reconstructed image is represented by u . Without any modification of the original Cahn–Hilliard equation, the image u would change drastically. In other words, it would have nothing in common with the original image f anymore. This means we have to include information about D and f in the model. Bertozzi, Esedoğlu, and Gillette [8, 7] considered a second energy functional

$$(2.7) \quad \mathcal{E}_2(u) = \int_{\Omega} \frac{\omega}{2} (f - u)^2 d\mathbf{x} = \int_{\Omega \setminus D} \frac{\omega_0}{2} (f - u)^2 d\mathbf{x}.$$

It keeps the solution u close to the given image f in the undamaged parts. The gradient flow of \mathcal{E}_2 with respect to the L^2 -inner product results in the fidelity term in (2.1).

Finally, Bertozzi, Esedoğlu, and Gillette’s [8, 7] black-and-white Cahn–Hilliard inpainting model (2.1)–(2.2) is given as a superposition of the H^{-1} -gradient flow for \mathcal{E}_1 and the L^2 -gradient flow for \mathcal{E}_2 . The evolution of u can be described as follows: Outside of the damaged regions, u stays close to the given image f . We can control this closeness via the fidelity parameter ω_0 . Based on this known image information, the damaged areas are filled in. The different gray level lines ending at the boundaries of the damaged parts smoothly continue inward. Inside these regions, the original Cahn–Hilliard equation acts. It maximizes the occurrence of black and white while simultaneously minimizing the curvature of their interface. Finally, the inpainted image is constructed by following this evolution to steady state.

Remark 2.1. The gradient flow of \mathcal{E}_1 with respect to the L^2 -inner product yields the Allen–Cahn equation [2]. Regarding the application of the Allen–Cahn model to inpainting we refer

the reader to Sarbu [60]. Moreover, Li et al. [50] recently proposed a local Allen–Cahn inpainting approach which acts only on the inpainting domain.

The effective application of the modified Cahn–Hilliard equation to binary images motivates us to study its natural generalization based on the vector-valued Cahn–Hilliard equation.

3. Gray value Cahn–Hilliard inpainting. We now formulate the gray value inpainting model based on the vector-valued Cahn–Hilliard equation [33, 30]. Let f be a given gray value image defined on Ω , let N be the number of desired gray values, and let $\mathbf{g} = (g_1, \dots, g_N)^T$ be the vector of desired gray values. Note that $2 \leq N \leq 256$. A vector-valued phase variable $\mathbf{u} = (u_1, \dots, u_N)^T: \Omega \times (0, T)$ is introduced. The component u_i describes the evolution of gray value g_i for $i = 1, \dots, N$. This means that if $u_i(\mathbf{x}, t) = 0$, then gray value g_i is absent in \mathbf{x} , and if $u_i(\mathbf{x}, t) = 1$, only gray value g_i is present in there. Hence

$$(3.1) \quad \sum_{i=1}^N u_i = 1$$

and $u_i \geq 0, i = 1, \dots, N$, is required, so that admissible states belong to the Gibbs simplex

$$(3.2) \quad \mathcal{G}^N := \left\{ \mathbf{v} \in \mathbb{R}^N \mid \sum_{i=1}^N v_i = 1, v_i \geq 0 \text{ for } i = 1, \dots, N \right\}.$$

As with Bertozzi, Esedoglu, and Gillette’s [8, 7] black-and-white Cahn–Hilliard inpainting model, our proposed model arises as a superposition of two gradient flows: a projected H^{-1} -gradient flow for the Cahn–Hilliard part and an L^2 -gradient flow for the fidelity term. The use of a projected gradient flow accounts for the additional constraint (3.1), which has to hold everywhere at any time. The generalization of the Ginzburg–Landau energy (2.4) to multicomponent systems is given as

$$(3.3) \quad \mathcal{E}_1(\mathbf{u}) = \int_{\Omega} \left\{ \frac{\varepsilon}{2} \sum_{i=1}^N |\nabla u_i|^2 + \frac{1}{\varepsilon} \psi(\mathbf{u}) \right\} \mathrm{d}\mathbf{x}.$$

As already mentioned in section 2.1, the present work concentrates on a smooth potential function ψ , and the well-known potential function (2.5) generalizes to

$$(3.4) \quad \psi(\mathbf{u}) = \frac{1}{4} \sum_{i=1}^N u_i^2 (1 - u_i)^2.$$

The vector-valued Cahn–Hilliard equation is the projected H^{-1} -gradient flow of the energy \mathcal{E}_1 in (3.3). By this we mean the H^{-1} -gradient flow of \mathcal{E}_1 under the additional constraint (3.1). In order to obtain the governing equations, we need to calculate the variational derivative of \mathcal{E}_1 , i.e.,

$$\left(\frac{\delta \mathcal{E}_1(\mathbf{u})}{\delta \mathbf{u}} \right) (\mathbf{v}) := \lim_{\eta \rightarrow 0} \frac{\mathcal{E}_1(\mathbf{u} + \eta \mathbf{v}) - \mathcal{E}_1(\mathbf{u})}{\eta}$$

for all sufficiently smooth functions $\mathbf{u}: \Omega \times (0, T) \rightarrow \mathcal{U}$ and $\mathbf{v}: \Omega \times (0, T) \rightarrow \mathcal{V}$, where

$$(3.5) \quad \mathcal{U} = \left\{ \mathbf{w} \in \mathbb{R}^N \mid \sum_{i=1}^N w_i = 1 \right\},$$

$$(3.6) \quad \mathcal{V} = \left\{ \mathbf{w} \in \mathbb{R}^N \mid \sum_{i=1}^N w_i = 0 \right\}.$$

In doing so, we choose a general smooth function $\boldsymbol{\xi}: \Omega \times (0, T) \rightarrow \mathbb{R}^N$ and set $\mathbf{v} = P\boldsymbol{\xi}$. The orthogonal projection $P\boldsymbol{\xi} = \boldsymbol{\xi} - \frac{1}{N}(\mathbf{1} \cdot \boldsymbol{\xi})\mathbf{1}$ maps \mathbb{R}^N onto the linear subspace \mathcal{V} in (3.6). As mentioned above, it accounts for the fact that admissible variations of $\mathbf{u} \in \mathcal{U}$ must be in \mathcal{U} . In doing so, we finally obtain the vector-valued Cahn–Hilliard equation

$$(3.7) \quad \partial_t \mathbf{u} = -\Delta \left(\varepsilon \Delta \mathbf{u} - \frac{1}{\varepsilon} P\psi'(\mathbf{u}) \right),$$

$$(3.8) \quad \nabla u_i \cdot \mathbf{n} = \nabla(\Delta u_i) \cdot \mathbf{n} = 0 \quad \text{on } \partial\Omega, \quad i = 1, \dots, N.$$

Here and in the following, we use the notation $\Delta \mathbf{u} = (\Delta u_1, \dots, \Delta u_N)^T$, $\mathbf{1} = (1, \dots, 1)^T \in \mathbb{R}^N$, and $\psi'(\mathbf{u}) = (\frac{\delta\psi}{\delta u_1}(\mathbf{u}), \dots, \frac{\delta\psi}{\delta u_N}(\mathbf{u}))$. For more details, we refer the reader to [37, 30, 48, 25, 47].

The generalization of the energy (2.7) for the fidelity term to multicomponent systems is given as

$$(3.9) \quad \mathcal{E}_2(\mathbf{u}) = \int_{\Omega} \left\{ \frac{\omega}{2} \sum_{i=1}^N (f_i - u_i)^2 \right\} \mathrm{d}\mathbf{x}.$$

Here, $\mathbf{f} = (f_1, \dots, f_N)^T \in \mathcal{G}^N$ is the vector of given gray value distributions from the original image f . That means that $f_i \in \{0, 1\}$ describes the intensity of the gray value g_i in f for $i = 1, \dots, N$.

Finally, our proposed generalization of the black-and-white Cahn–Hilliard inpainting model (2.1)–(2.2) to gray value images is given as a superposition of the projected H^{-1} -gradient flow for \mathcal{E}_1 in (3.3) and the L^2 -gradient flow for \mathcal{E}_2 in (3.9), i.e.,

$$\partial_t \mathbf{u} = -\nabla_{H^{-1}}^{\text{proj}} \mathcal{E}_1(\mathbf{u}) - \nabla_{L^2} \mathcal{E}_2(\mathbf{u}).$$

Here, ∇_{L^2} represents the gradient descent with respect to the L^2 -inner product. The term $\nabla_{H^{-1}}^{\text{proj}}$ represents the projected gradient descent with respect to the H^{-1} -inner product. This reads componentwise as

$$(3.10) \quad \partial_t u_i = -\Delta \left(\varepsilon \Delta u_i - \frac{1}{\varepsilon} \frac{\partial \psi}{\partial u_i}(\mathbf{u}) + \frac{1}{\varepsilon N} \sum_{i=1}^N \frac{\partial \psi}{\partial u_i}(\mathbf{u}) \right) + \omega(f_i - u_i),$$

$$(3.11) \quad \nabla u_i \cdot \mathbf{n} = \nabla(\Delta u_i) \cdot \mathbf{n} = 0 \quad \text{on } \partial\Omega$$

for $i = 1, \dots, N$.

Remark 3.1. Note that (3.10)–(3.11) is still a simplified vector-valued model. In fact, for many applications with a focus on phase separation and coarsening processes, a concentration-dependent mobility matrix $L = L(\mathbf{u}) = (L_{ij}(\mathbf{u}))_{i,j=1,\dots,N}$ is required in the corresponding

model. This is the case, for example, if the mobility in the interface is larger than in the pure phases. Nevertheless, Lee, Choi, and Kim [47] show reasonable results with the simplified choice $L = I$, where I is the identity matrix. For this reason, and because the task of inpainting is to get a final result as fast as possible (and not to accurately develop the evolution of the phases), we omit the mobility matrix in our model.

Remark 3.2. Recently, Cherfils, Fakhri, and Miranville [25] considered and analyzed a similar model. More precisely, they proved the existence and uniqueness of solutions, as well as the existence of the global attractor. Moreover, they constructed finite-dimensional attractors and proved that their model is algebraically consistent with the two-phase model. In contrast, our work focuses on the numerical analysis of an unconditionally time-stepping scheme as well as its efficient numerical solution. Further, we generalize our approach to a fractional-in-space version.

Remark 3.3. Note that a generalized Ginzburg–Landau energy functional as in (3.3) has been used in other contexts in phase field methods. In particular, in [36, 51, 52, 9] the authors studied problems in the classification of high-dimensional data. Merkurjev, Kostić, and Bertozzi [52] and Bertozzi and Flenner [9] started with binary data classification based on the Ginzburg–Landau energy functional in (2.4). Especially, Merkurjev et al. applied their semi-supervised image classification algorithm to inpainting. Recently, Garcia-Cardona et al. [36] and Merkurjev et al. [51] extended the binary classification algorithm to a multiclass method using the idea of the Gibbs simplex (3.2). All the works above used a generalized modified Allen–Cahn equation which is defined for functions on graphs. In particular, their approaches involve extensions of the classical Laplace operator to a more general graph Laplacian. This is in the same spirit as [28], in which the authors generalized the Ginzburg–Landau functional to wavelets. In this sense, our proposed extension of the classical Laplace operator to the fractional Laplacian (see section 5) can also be categorized as a similar approach. In contrast to our method, the approach in [36, 51] is not based on a projected gradient flow. Instead, the authors use the procedure in [24] to project the phase field back to the Gibbs simplex.

4. Convexity splitting. In the case of black-and-white Cahn–Hilliard inpainting, Bertozzi, Esedoğlu, and Gillette [8] proposed a semi-implicit scheme, the convexity splitting scheme. The authors conjectured unconditionally stability in the sense that solutions of the numerical scheme are bounded within a finite time interval, independent of the time step size. Indeed, Schönlieb and Bertozzi [61] proved consistency, unconditional stability, and convergence of this scheme.

The convexity splitting method was originally introduced by Elliott and Stuart [31] and is often attributed to Eyre [34]. Actually, it was designed to solve gradient systems. But it can also be applied in a modified form to evolution equations that do not follow a variational principle. In particular, such equations include the Cahn–Hilliard inpainting models (2.1)–(2.2) and (3.10)–(3.11), as described further on.

In the following, we extend the numerical analysis of the convexity splitting scheme for the scalar inpainting model (2.1)–(2.2) studied in [61] to the vector-valued inpainting model (3.10)–(3.11).

The original idea of convexity splitting applied to gradient systems is to write the considered energy functional as the sum of a convex plus a concave energy functional. The

convex part is then treated implicitly, while the concave part is treated explicitly. Under the right conditions, this approach leads to an unconditionally gradient stable time-discretization scheme.

As pointed out in the previous sections, the modified Cahn–Hilliard equation as a whole is not given by a gradient flow. Especially, our proposed model arises as a superposition of the projected H^{-1} -gradient flow for \mathcal{E}_1 in (3.9) and the L^2 -gradient flow for \mathcal{E}_2 in (3.9). In this case, convexity splitting is applied to each of these energies. To be more precise, we split \mathcal{E}_1 as $\mathcal{E}_1 = \mathcal{E}_{1c} - \mathcal{E}_{1e}$, where

$$\begin{aligned}\mathcal{E}_{1c}(\mathbf{u}) &= \int_{\Omega} \left\{ \frac{\varepsilon}{2} \sum_{i=1}^N |\nabla u_i|^2 + \frac{C_1}{2} \sum_{i=1}^N u_i^2 \right\} dx, \\ \mathcal{E}_{1e}(\mathbf{u}) &= \int_{\Omega} \left\{ -\frac{1}{\varepsilon} \psi(\mathbf{u}) + \frac{C_1}{2} \sum_{i=1}^N u_i^2 \right\} dx,\end{aligned}$$

as well as $\mathcal{E}_2 = \mathcal{E}_{2c} - \mathcal{E}_{2e}$, where

$$\begin{aligned}\mathcal{E}_{2c}(\mathbf{u}) &= \int_{\Omega} \left\{ \frac{C_2}{2} \sum_{i=1}^N u_i^2 \right\} dx, \\ \mathcal{E}_{2e}(\mathbf{u}) &= \int_{\Omega} \left\{ -\frac{\omega}{2} \sum_{i=1}^N (f_i - u_i)^2 + \frac{C_2}{2} \sum_{i=1}^N u_i^2 \right\} dx.\end{aligned}$$

The constants C_1 and C_2 are positive and need to be chosen large enough such that the energies \mathcal{E}_{1c} , \mathcal{E}_{1e} , \mathcal{E}_{2c} , and \mathcal{E}_{2e} are strictly convex. That means that C_1 has to be comparable to $\frac{1}{\varepsilon}$ and C_2 has to be comparable to ω_0 . These convexity requirements are the same as for the black-and-white inpainting model. Also, the proof is similar, and we refer the reader to Appendix A in the supplementary materials (see M101405_01.pdf [local/web 98.8KB]).

The resulting discrete time-stepping scheme is given by

$$\frac{\mathbf{U}^{(n)} - \mathbf{U}^{(n-1)}}{\tau} = -\nabla_{H^{-1}}^{\text{proj}} \left(\mathcal{E}_{1c}(\mathbf{U}^{(n)}) - \mathcal{E}_{1e}(\mathbf{U}^{(n-1)}) \right) - \nabla_{L^2} \left(\mathcal{E}_{2c}(\mathbf{U}^{(n)}) - \mathcal{E}_{2e}(\mathbf{U}^{(n-1)}) \right).$$

Here, $\tau > 0$ denotes the time step size and $n \in \mathbb{N}$ the time step. This translates to a numerical scheme of the form

$$\begin{aligned}(4.1) \quad & \frac{U_i^{(n)} - U_i^{(n-1)}}{\tau} + \varepsilon \Delta^2 U_i^{(n)} - C_1 \Delta U_i^{(n)} + C_2 U_i^{(n)} \\ &= \frac{1}{\varepsilon} \Delta \left(\frac{\partial \psi}{\partial u_i}(\mathbf{U}^{(n-1)}) \right) - \frac{1}{\varepsilon N} \Delta \left(\sum_{j=1}^N \frac{\partial \psi}{\partial u_j}(\mathbf{U}^{(n-1)}) \right) + \omega (f_i - U_i^{(n-1)}) - C_1 \Delta U_i^{(n-1)} + C_2 U_i^{(n-1)},\end{aligned}$$

$$(4.2) \quad \nabla U_i \cdot \mathbf{n} = \nabla(\Delta U_i) \cdot \mathbf{n} = 0 \quad \text{on } \partial\Omega$$

for $i = 1, \dots, N$. Following [61, pp. 425–434], we have proved consistency, unconditional stability, and convergence of the scheme (4.1)–(4.2). For the proof, we refer the reader to Appendix B in the supplementary materials (see M101405_01.pdf [local/web 98.8KB]).

5. Fourier spectral methods. Bertozzi, Esedoğlu, and Gillette [8] benefited from the use of the FFT to achieve fast inpainting based on the scalar Cahn–Hilliard equation. They proposed a two-dimensional FFT method to compute the finite differences for the derivatives. We adopt a slightly different approach and exploit the spectral decomposition of the occurring Laplace operator. That means we can still make use of the FFT but not of the finite differences. We emphasize the well-known symmetry-preservation properties of spectral methods for the Laplacian operator. Lower order methods, such as finite difference methods, have a tendency to form squares from axisymmetric initial conditions. The idea of our approach comes from the work of Bueno-Orovio, Kay, and Burrage [14], who introduced Fourier spectral methods for fractional-in-space reaction-diffusion equations. Thus, we will present this approach for a fractional version of the Cahn–Hilliard inpainting model in the following. In doing so, we reformulate (4.1)–(4.2) to a more general form by replacing the standard Laplace operator, Δ , by its fractional counterpart, $-(-\Delta)^{\frac{\zeta}{2}}$:

$$\begin{aligned}
 (5.1) \quad & \frac{U_i^{(n)} - U_i^{(n-1)}}{\tau} + \varepsilon \Delta^\zeta U_i^{(n)} + C_1 (-\Delta)^{\frac{\zeta}{2}} U_i^{(n)} + C_2 U_i^{(n)} \\
 & = -\frac{1}{\varepsilon} (-\Delta)^{\frac{\zeta}{2}} \left(\frac{\partial \psi}{\partial u_i}(\mathbf{U}^{(n-1)}) \right) + \frac{1}{\varepsilon N} (-\Delta)^{\frac{\zeta}{2}} \left(\sum_{j=1}^N \frac{\partial \psi}{\partial u_j}(\mathbf{U}^{(n-1)}) \right) \\
 & \quad + \omega(f_i - U_i^{(n-1)}) + C_1 (-\Delta)^{\frac{\zeta}{2}} U_i^{(n-1)} + C_2 U_i^{(n-1)},
 \end{aligned}$$

$$(5.2) \quad \nabla U_i \cdot \mathbf{n} = \nabla (\Delta U_i) \cdot \mathbf{n} = 0 \quad \text{on } \partial\Omega.$$

Here, ζ denotes the fractional power which represents superdiffusion for $0 < \zeta < 2$ (fractional Laplacian) and pure diffusion for $\zeta = 2$ (standard Laplacian).

As pointed out in [54, 66], there is no unique way to define the fractional Laplacian in a bounded domain. One possibility is based on the Fourier transform on an infinite domain [59]. It has a natural extension to include finite domains when the function is subject to homogeneous Dirichlet boundary conditions. Another definition uses the eigenfunction expansion on a finite domain [42]. As already mentioned above, this is the interpretation we adopt. Let us now summarize the main elements of the spectral approach according to [14] for the two-dimensional case. Three spatial dimensions can be constructed straightforwardly.

It is known that the Laplacian $(-\Delta)$ has a complete set of orthonormal eigenfunctions $\{\varphi_{\alpha,\beta}\}$ satisfying the Neumann boundary conditions on a bounded region $\Omega \subset \mathbb{R}^d$ with corresponding eigenvalues $\lambda_{\alpha,\beta}$; see also [58, Theorem 8.6]. More precisely, we consider the following eigenvalue problem in $\Omega = [0, a] \times [0, b]$:

$$\begin{aligned}
 (-\Delta)\varphi_{\alpha,\beta} &= \lambda_{\alpha,\beta}\varphi_{\alpha,\beta}, \\
 \nabla\varphi_{\alpha,\beta} \cdot \mathbf{n} &= 0 \quad \text{on } \partial\Omega,
 \end{aligned}$$

$\alpha, \beta = 1, 2, \dots$. Then we have

$$\lambda_{\alpha, \beta} = \pi^2 \left(\frac{(\alpha - 1)^2}{a^2} + \frac{(\beta - 1)^2}{b^2} \right),$$

$$\varphi_{\alpha, \beta} = \frac{2}{\sqrt{ab}} \cos \left(\frac{(\alpha - 1)\pi x}{a} \right) \cos \left(\frac{(\beta - 1)\pi y}{b} \right).$$

Following [14], we define

$$(5.3) \quad \mathcal{U}_\zeta := \left\{ u = \sum_{\alpha=1}^{\infty} \sum_{\beta=1}^{\infty} \hat{u}(\alpha, \beta) \varphi_{\alpha, \beta} \in L^2(\Omega), \hat{u}(\alpha, \beta) = (u, \varphi_{\alpha, \beta}), \right. \\ \left. \sum_{\alpha=1}^{\infty} \sum_{\beta=1}^{\infty} |\hat{u}(\alpha, \beta)|^2 |\lambda_{\alpha, \beta}|^{\frac{\zeta}{2}} < \infty, 0 < \zeta \leq 2 \right\},$$

where (\cdot, \cdot) stands for the $L^2(\Omega)$ -inner product. Then, for any $u \in \mathcal{U}_\zeta$, the fractional Laplace operator can be defined via

$$(5.4) \quad (-\Delta)^{\frac{\zeta}{2}} u = \sum_{\alpha=1}^{\infty} \sum_{\beta=1}^{\infty} \hat{u}(\alpha, \beta) \lambda_{\alpha, \beta}^{\frac{\zeta}{2}} \varphi_{\alpha, \beta},$$

where $\hat{u}(\alpha, \beta)$ denote the Fourier coefficients of u . The basic idea of Fourier spectral methods is that smooth functions have a rapidly decaying transform. Fourier spectral methods represent the truncated series expansion when a finite number of orthonormal eigenfunction $\{\varphi_{\alpha, \beta}\}$ is considered.

Remark 5.1. Note that we have proved the consistency, unconditional stability, and convergence only for the nonfractional model (4.1)–(4.2). An extension of the numerical analysis to the fractional-order scheme (5.1)–(5.2) is a topic of future research. However, in section 6 we numerically demonstrate the convergence of the fractional model.

We now present the mesh discretization. We denote by m_x and m_y the number of internal equispaced mesh points in the x - and y -directions, hence not including boundary nodes. The mesh points are given by

$$x_{k,l} = \left((l - 1)h_x + \frac{h_x}{2}, (k - 1)h_y + \frac{h_y}{2} \right), \quad l = 1, \dots, m_x, k = 1, \dots, m_y,$$

where $h_x = \frac{1}{m_x}$ and $h_y = \frac{1}{m_y}$ denote the mesh sizes in both dimensions. The discrete versions of the unknown functions $u_i^{(n)}$, $i = 1, \dots, N$, in (4.1)–(4.2) are written in matrix form as

$$U_i^{(n)} = \begin{bmatrix} U_i^{(n)}(1, 1) & \cdots & U_i^{(n)}(1, m_x) \\ \vdots & \ddots & \vdots \\ U_i^{(n)}(m_y, 1) & \cdots & U_i^{(n)}(m_y, m_x) \end{bmatrix},$$

where $U_i^{(n)}(k, l)$ is the approximation of the function $u_i^{(n)}$ at the mesh point $x_{k,l}$ for $k = 1, \dots, m_y, l = 1, \dots, m_x$. We denote by $\hat{U}_i^{(n)}$ the two-dimensional discrete cosine transform (DCT) of $U_i^{(n)}$

$$\hat{U}_i^{(n)} = \begin{bmatrix} \hat{U}_i^{(n)}(1, 1) & \cdots & \hat{U}_i^{(n)}(1, m_x) \\ \vdots & \ddots & \vdots \\ \hat{U}_i^{(n)}(m_y, 1) & \cdots & \hat{U}_i^{(n)}(m_y, m_x) \end{bmatrix}$$

for $i = 1, \dots, N$. In the same manner of capital letters, we define the discrete versions of f_i . Applying the Fourier transform to both sides of (5.1) and using the spectral decomposition of the Laplacian (5.4), we get

$$(5.5) \quad \hat{U}_i^{(n)}(\alpha, \beta) = \frac{\lambda_{\alpha, \beta}^{\frac{\zeta}{2}} \left[\left(-\frac{1}{\varepsilon} \frac{\partial \psi}{\partial u_i}(\mathbf{U}^{(n-1)}) + \frac{1}{\varepsilon N} \sum_{j=1}^N \frac{\partial \psi}{\partial u_j}(\mathbf{U}^{(n-1)}) \right)^\wedge(\alpha, \beta) + C_1 \hat{U}_i^{(n-1)}(\alpha, \beta) \right]}{\frac{1}{\tau} + \varepsilon \lambda_{\alpha, \beta}^\zeta + C_1 \lambda_{\alpha, \beta}^{\frac{\zeta}{2}} + C_2} + \frac{\left[\omega F_i + (C_2 - \omega) U_i^{(n-1)} \right]^\wedge(\alpha, \beta) + \frac{1}{\tau} \hat{U}_i^{(n-1)}(\alpha, \beta)}{\frac{1}{\tau} + \varepsilon \lambda_{\alpha, \beta}^\zeta + C_1 \lambda_{\alpha, \beta}^{\frac{\zeta}{2}} + C_2}$$

for $i = 1, \dots, N$ and $\alpha = 1, \dots, m_y, \beta = 1, \dots, m_x$. The system (5.5) is fully diagonal and the two-dimensional DCT matrices $\hat{U}_i^{(n)}, i = 1, \dots, N$, can be computed rapidly and effectively. All that remains is to compute the inverse two-dimensional DCT on each $\hat{U}_i^{(n)}$ in order to get the discrete solutions $U_i^{(n)}$. Note that the computational costs per time step are the same for the standard and fractional Cahn–Hilliard inpainting model.

6. Numerical results. In this section, we present numerical results for the models. For all simulations we use Bertozzi, Esedoğlu, and Gillette’s [8] ε -two-step approach. This procedure successfully connects edges across large inpainting regions. In the first step, we run the Cahn–Hilliard inpainting approach close to steady state with a rather large value of ε . In the second step, the approximate solution from the first step serves as the initial state for a second run of Cahn–Hilliard inpainting. But this time we set ε to a small value. In summary, the first step smooths the image information. Hence, level lines can merge over large damaged regions. The second step sharpens the image contours. In both rounds, the stopping criterion is

$$(6.1) \quad \frac{\|\mathbf{U}^{(n)} - \mathbf{U}^{(n-1)}\|_2}{\|\mathbf{U}^{(n-1)}\|_2} \leq \epsilon$$

if not mentioned otherwise. We choose $\epsilon = 2 \cdot 10^{-4}$ for the first step and $\epsilon = 2 \cdot 10^{-5}$ for the second. In section 6.3, we compare (6.1) with other different stopping criteria.

We set the convexity parameters to $C_1 = \frac{3}{\varepsilon}, C_2 = 3\omega_0$ and the time step size to $\tau = 1$. Remember, ω_0 defined in (2.3) is the fidelity parameter that keeps the inpainted image close enough to the original picture. The initialization of the N phase variables is done in two preprocessing steps. First, we segment N clusters using the standard k-means clustering

method. The obtained cluster centroid locations serve as the gray values g_i , $i = 1, \dots, N$. Each phase variable represents one cluster and hence describes the evolution of one gray value. That means every phase variable is set to be one in its corresponding cluster region, and zero everywhere else. Second, we set every phase variable in the damaged regions to the value $\frac{1}{N}$. This assignment fulfills the conditions of the Gibbs simplex (3.2). The final image u_{inpaint} is obtained by

$$(6.2) \quad u_{\text{inpaint}} = \sum_{i=1}^N g_i U_i^{(n)},$$

where $\mathbf{U}^{(n)}$ represents the final phase variable. In section 6.3, we compare the merging technique (6.2) with the thresholding technique used in [25].

We use the PSNR

$$\text{PSNR} = 20 \log_{10} \left(\frac{1}{\sqrt{\frac{1}{m_x m_y} \sum_{i=1}^{m_y} \sum_{j=1}^{m_x} (u_{\text{orig}}(i, j) - u_{\text{inpaint}}(i, j))^2}} \right)$$

as well as the SSIM [65] to measure the quality of reconstruction.³ Here, u_{orig} denotes the original image without damaged regions. Traditional quality measures like the PSNR are not very well matched to perceived visual quality. In contrast, the SSIM takes advantage of known characteristics of the human visual system.

All computations are executed in MATLAB R2012b on a 64-bit server with CPU type Intel Xeon X5650 @2.67 GHz, with 2 CPUs, 12 cores (6 cores per CPU), and 48 GB main memory available. Note that we use the MATLAB Image Processing Toolbox for combining the phases into the final image and for visualizations.

We provide the MATLAB code to reproduce the numerical examples as supplementary material with this paper (see M101405_02.zip [local/web 1.71MB]).

6.1. Comparison to previous inpainting methods. In this section, we show the performance of various inpainting methods. Besides the proposed vector-valued Cahn–Hilliard (vector CH) inpainting model we test the bitwise binary inpainting approach (bitwise CH) as well as the MATLAB function `inpaintn`⁴ [35, 63].

Schönlieb provides three inpainting codes.⁵ These are inpainting methods using the heat equation (heat), TV inpainting, and TV- H^{-1} inpainting. We fix the time step and mesh sizes to $\tau = 0.1$, $h_x = h_y = 1$ in the heat equation based model and to $\tau = 1$, $h_x = h_y = 1$ in both TV inpainting approaches. Finally, we test Zhou et al's [70] nonparametric Bayesian method, which they term the beta process factor analysis (BPFA).⁶ In all examples below, we run BPFA with the default values.

³A code for calculating the SSIM index is available at <http://www.mathworks.com/matlabcentral/fileexchange/42238-an-edge-adaptive-directional-total-variation-model>.

⁴<http://www.mathworks.com/matlabcentral/fileexchange/27994-inpaint-over-missing-data-in-1-d--2-d--3-d--n-d-arrays>.

⁵<http://www.mathworks.com/matlabcentral/fileexchange/34356-higher-order-total-variation-inpainting>.

⁶<http://people.duke.edu/~mz31/Results/BPFAImage/>.

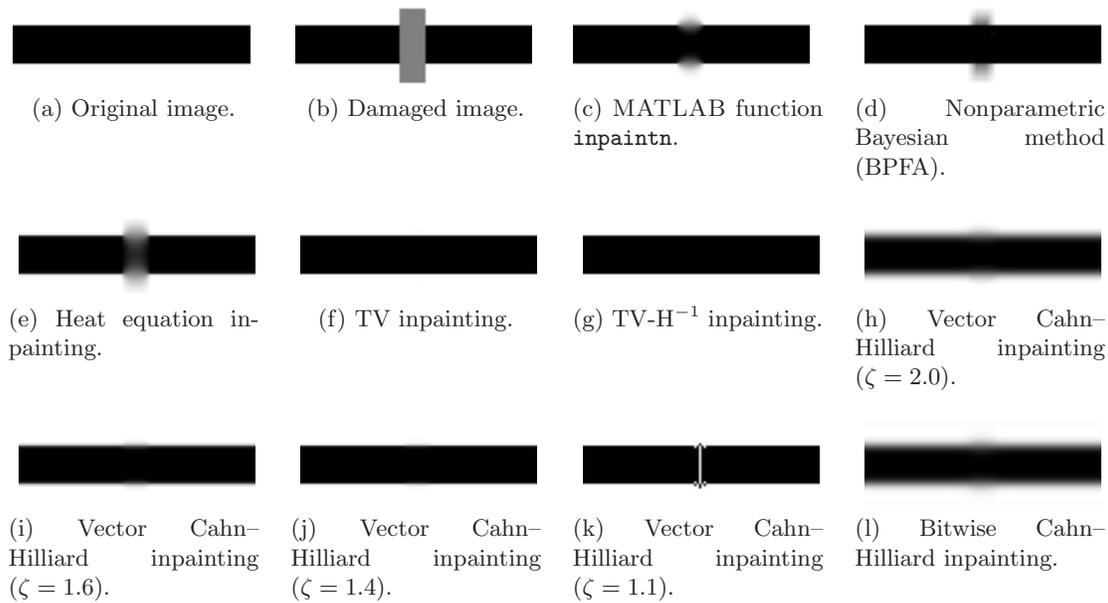


Figure 1. Inpainted black-and-white stripe image using different inpainting models.

In contrast to our proposed vector-valued model, the methods to which we compare it represent scalar systems. Hence, we do not need to segment the image for their initialization. By experience from those models, we set the pixel values in the damaged regions to 0 instead of $\frac{1}{N}$. Note that the gray values range from 0 to 255 for bitwise CH. For the remaining methods, the range is from 0 to 1.

The first test example is a simple 50×100 binary image containing a black stripe; see Figure 1(a). Figure 1(b) shows the damaged version. Figures 1(c)–1(l) illustrates the inpainted images using the different approaches mentioned above. Table 1 contains the total number of iterations, the total computational times (in seconds), the average computational times per time step (in seconds), the PSNR and SSIM values, the minimum and maximum pixel values of the final image, as well as the numbers of grid points within the interface in $\Omega \setminus D$ (in D) for the phase field models. In all Cahn–Hilliard approaches, we set $\omega_0 = 10^5$, $h_x = h_y = \frac{1}{100}$ and apply the ε -two-step procedure with a switch after 100 iterations from $\varepsilon = 1$ to $\varepsilon = h_x$. The fractional Cahn–Hilliard model is applied with fractional powers of $\zeta = 1.6, 1.4, 1.1$. We run the heat equation–based approach with $\omega_0 = 10$ and both TV models with $\omega_0 = 10$, $\varepsilon = \frac{1}{100}$. Except for the BPFA, we stop all methods at time step 10^5 .

The best results are obtained by both TV models. However, the fractional Cahn–Hilliard inpainting approach can compete with them in terms of image quality. We observe a significant improvement of our proposed model, when we reduce the fractional power from 2 to 1.6. The image contours are sharpened, and the PSNR and SSIM values are enhanced by a factor of 1.4 and 1.5. The transition from $\zeta = 2.0$ to $\zeta = 1.4$ is even better. The PSNR and SSIM values are enhanced by a factor of 1.9 and 1.6. Going further down to a fractional power of 1.1 deteriorates the results. Spurious artifacts occur across the interface. This is due to the

Table 1

Results for the black-and-white stripe inpainting using different inpainting models: The total number of iterations (*Iter.*), total computational times in seconds (*CPU*), average computational times per time step in seconds ($\overline{\text{CPU}}$), PSNR and SSIM values, minimum (*min*) and maximum (*max*) pixel values of the final image, as well as the numbers of grid points within the interface in $\Omega \setminus D$ (in *D*) for the phase field models.

Inpainting method	Iter.	CPU (s)	$\overline{\text{CPU}}$ (s)	PSNR	SSIM	min	max	Interface
<code>inpaintn</code>	100000	169.31	0.0017	27.37	0.9480	$-4.23 \cdot 10^{-2}$	1.0015	-
BPFA	1072	246.91	0.2303	26.67	0.9247	$-1.28 \cdot 10^{-1}$	1.0009	-
heat	1000000	110.91	0.0001	23.65	0.9010	$2.46 \cdot 10^{-9}$	1.0000	-
TV	100000	158.93	0.0016	59.76	0.9975	$1.51 \cdot 10^{-5}$	1.0000	-
TV-H ⁻¹	100000	162.43	0.0016	69.92	0.9999	$-4.01 \cdot 10^{-5}$	1.0000	-
vector CH ($\zeta = 2.0$)	100000	743.16	0.0074	18.30	0.6220	$-2.25 \cdot 10^{-2}$	1.0187	6 (7)
vector CH ($\zeta = 1.6$)	100000	830.96	0.0083	25.54	0.9499	$-7.01 \cdot 10^{-3}$	1.0069	2 (4)
vector CH ($\zeta = 1.4$)	100000	1046.90	0.0105	34.60	0.9934	$-1.07 \cdot 10^{-2}$	1.0108	2 (2)
vector CH ($\zeta = 1.1$)	100000	949.76	0.0095	23.58	0.9547	$-9.16 \cdot 10^{-4}$	1.0009	0 (-)
bitwise CH	100000	3612.57	0.0361	18.30	0.6220	$-5.73 \cdot 10^{+0}$	259.78	6 (7)

fact that the interface is no longer resolved fine enough. Table 1 shows that there is no mesh point across the interface. We further elaborate on the influence of the fractional power in section 6.2.

Comparing the computational times between the TV models and the vector-valued approach, we are at a disadvantage. The CPU time increases by a factor of five. Bitwise binary inpainting further increases the computational time since this approach is in terms of costs comparable to the 8-component Cahn–Hilliard inpainting method. Note that since we only deal with a black-white image here, a vector-valued or bitwise approach is not the method of choice. But this experiment shows that the results of these models are in accordance with the corresponding scalar models.

The second test example consists of six stripes spanning different widths and is of size 64×64 ; see Figure 2(a). It is composed of five gray values. Figure 2(b) shows the damaged version. The Figures 2(c)–2(l) illustrate the inpainted images using the different approaches mentioned above. Table 2 lists the total number of iterations, the total computational times (in seconds), the average computational times per time step (in seconds), the PSNR and SSIM values, the minimum and maximum pixel values of the final image, as well as the numbers of grid points within the interface in $\Omega \setminus D$ (in *D*) for the phase field models. In all Cahn–Hilliard approaches, we set $\omega_0 = 10^5$, $h_x = h_y = \frac{1}{64}$ and apply the ε -two-step procedure with a switch from $\varepsilon = 1$ to $\varepsilon = h_x$. The fractional Cahn–Hilliard model is performed with fractional powers of $\zeta = 1.6, 1.4, 1.1$. We run the heat equation–based approach with $\omega_0 = 10$ and both TV models with $\omega_0 = 10$, $\varepsilon = \frac{1}{64}$. We apply to these three methods the stopping criterion (6.1) with $\epsilon = 2 \cdot 10^{-5}$. We stop the MATLAB function `inpaintn` after 10^5 iterations.

Compared to the previous example in Figure 1(b), we have increased the inpainting gap. The second-order TV inpainting approach is no longer able to connect the stripes. Also BPFA fails using the standard parameter set. TV-H⁻¹ inpainting results in a partly complete connection. The two rightmost stripes have successfully joined. We observe the same after bitwise binary Cahn–Hilliard inpainting. In both cases, the stopping criterion (6.1) might not be the optimal choice. In general, the discussion about the stopping criterion should be a task

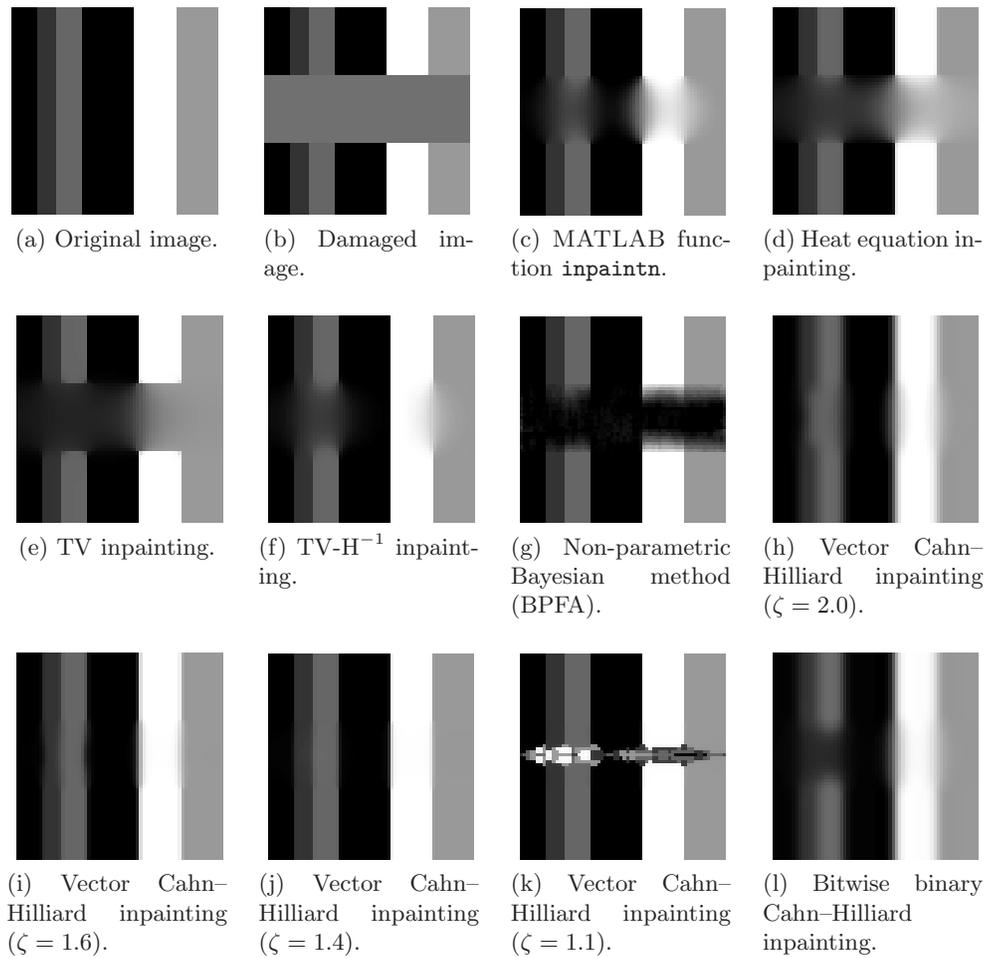


Figure 2. Inpainted gray value stripe image using different inpainting models.

for future work. In section 6.3, we compare different stopping criteria for our proposed model.

Both the nonfractional and fractional Cahn–Hilliard inpainting model provide a complete connection of the stripes over the inpainting domain. Again, our proposed fractional Cahn–Hilliard inpainting approach outperforms its nonfractional version. The PSNR and SSIM values are improved by a factor of about 1.3 when the fractional power is decreased from 2 to 1.6. A decrease from 2 to 1.4 improves the PSNR and SSIM values by a factor of 1.6 and 1.4. Again, spurious artifacts occur if the fractional model is no longer able to resolve the interface well enough, as can be seen in Figure 2(k).

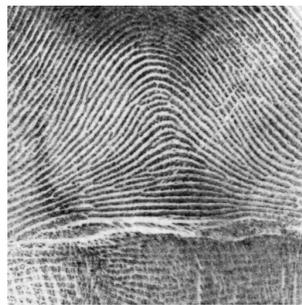
The third test example is a 512×512 fingerprint image;⁷ see Figure 3(a). Figure 3(b) displays an extract of the image after k-means clustering with 10 gray values. It zooms in the middle part of the fingerprint. This portion is taken for visual comparisons of the different inpainting methods. The damaged version of the whole image is illustrated in Figure 3(c).

⁷The image is taken from <http://people.duke.edu/~mz31/Results/BPFAImage/>.

Table 2

Results for the gray value stripe inpainting using different inpainting models: The total number of iterations (*Iter.*), total computational times in seconds (*CPU*), average computational times per time step in seconds ($\overline{\text{CPU}}$), PSNR and SSIM values, minimum (*min*) and maximum (*max*) pixel values of the final image, as well as the numbers of grid points within the interface in $\Omega \setminus D$ (*in D*) for the phase field models.

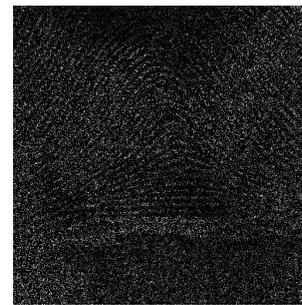
Inpainting method	Iter.	CPU (s)	$\overline{\text{CPU}}$ (s)	PSNR	SSIM	min	max	Interface
<code>inpaintn</code>	100000	234.17	0.0023	22.39	0.7839	$-1.05 \cdot 10^{-3}$	1.0000	-
BPFA	1072	143.48	0.1338	12.16	0.6618	$-1.14 \cdot 10^{-1}$	1.0010	-
heat	7248	6.42	0.0009	18.89	0.7010	$4.92 \cdot 10^{-10}$	1.0000	-
TV	27041	50.90	0.0019	16.51	0.6427	$5.54 \cdot 10^{-5}$	0.9999	-
TV- H^{-1}	85547	163.55	0.0019	26.97	0.8268	$-1.78 \cdot 10^{-3}$	1.0244	-
vector CH ($\zeta = 2.0$)	1404	28.23	0.0201	21.30	0.6846	$-2.34 \cdot 10^{-2}$	1.0241	4 (8)
vector CH ($\zeta = 1.6$)	1402	28.23	0.0201	27.24	0.9192	$-2.15 \cdot 10^{-2}$	1.0215	2 (4)
vector CH ($\zeta = 1.4$)	2499	50.48	0.0202	33.62	0.9664	$-1.08 \cdot 10^{-2}$	1.0109	0 (2)
vector CH ($\zeta = 1.1$)	18402	371.67	0.0202	17.29	0.8180	$-6.57 \cdot 10^{-4}$	1.0007	0 (-)
bitwise CH	1112	34.81	0.0313	20.54	0.6662	$-5.98 \cdot 10^{+0}$	261.15	4 (8)



(a) Original image.



(b) Clustered extract of the original image using 10 gray values.



(c) Damaged image.

Figure 3. Fingerprint image.

We have removed 80% of the pixels at random.

Figure 4 illustrates the inpainted (extracted) images using the different approaches mentioned above. The middle column of Table 3 lists the total number of iterations, the total computational times (in seconds), as well as the PSNR and SSIM values. In all Cahn–Hilliard approaches, we set $\omega_0 = 5 \cdot 10^8$, $h_x = h_y = \frac{1}{512}$ and apply the ε -two-step procedure with a switch from $\varepsilon = 1$ to $\varepsilon = h_x$. The fractional Cahn–Hilliard model is performed with a fractional power of $\zeta = 1.8$. We run the heat equation–based model with $\omega_0 = 10$, TV inpainting with $\omega_0 = 10^2$, $\varepsilon = \frac{1}{512}$, and TV- H^{-1} inpainting with $\omega_0 = 10^3$, $\varepsilon = \frac{1}{512}$. We apply to these three methods the stopping criterion (6.1) with $\epsilon = 2 \cdot 10^{-5}$. We stop the MATLAB function `inpaintn` after 10^4 iterations.

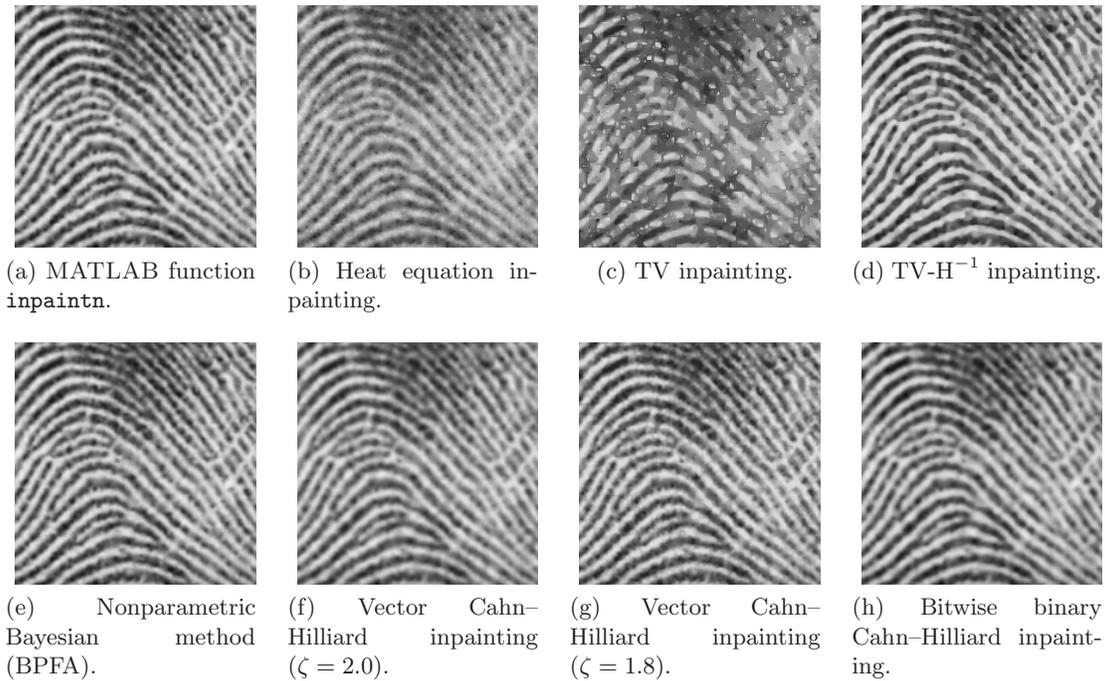


Figure 4. Inpainted fingerprint image using different inpainting models.

Table 3

Results for the fingerprint and hill inpainting: The total number of iterations (*Iter.*), computational times in seconds (*CPU*), and *PSNR/SSIM* values using different inpainting models. The labeling * denotes that the corresponding method did not converge to the tolerance $\epsilon = 2 \cdot 10^{-5}$ within the given number of iterations using the stopping criterion (6.1).

Inpainting method	Fingerprint image				Hill image			
	Iter.	CPU (s)	PSNR	SSIM	Iter.	CPU (s)	PSNR	SSIM
<code>inpaintn</code>	10000	483.63	26.30	0.8993	60000	2721.47	32.64	0.9583
BPFA	1072	9379.18	25.97	0.8850	1072	15941.66	21.75	0.9076
heat	787	121.00	21.82	0.7535	4300	644.65	32.35	0.9459
TV	10000*	2077.88	18.15	0.5720	60000*	12196.37	30.74	0.9473
TV- H^{-1}	10000*	2103.79	25.22	0.8783	60000*	12400.40	25.46	0.9344
vector CH ($\zeta = 2.0$)	503	1997.56	23.62	0.8162	2131	12402.95	29.94	0.8697
vector CH ($\zeta = 1.8$)	1438	5768.40	24.65	0.8615	2175	12057.60	31.18	0.9097
bitwise CH	338	1079.57	24.20	0.8326	915	2686.43	30.21	0.8838

Note that TV and TV- H^{-1} did not converge to the tolerance $\epsilon = 2 \cdot 10^{-5}$ within 10^4 iterations. Due to that, the inpainted fingerprints in Figure 4(c) and 4(d) look unsatisfactory. If we had continued the inpainting process, the results would have been of better quality, but at the expense of computational costs.

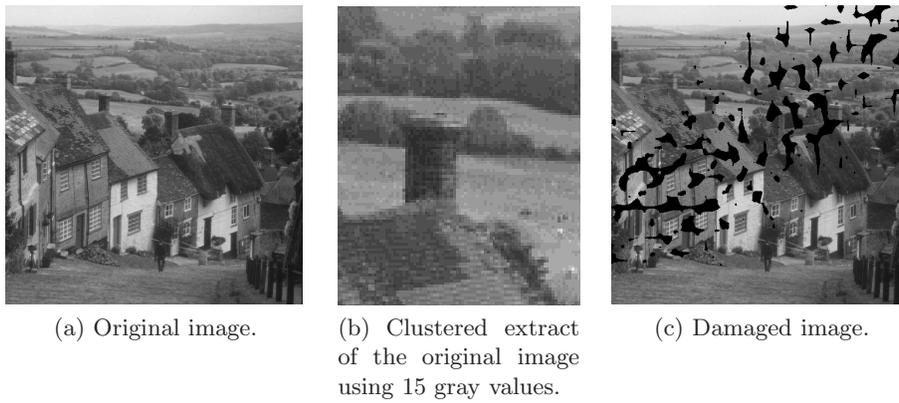


Figure 5. *Hill image.*

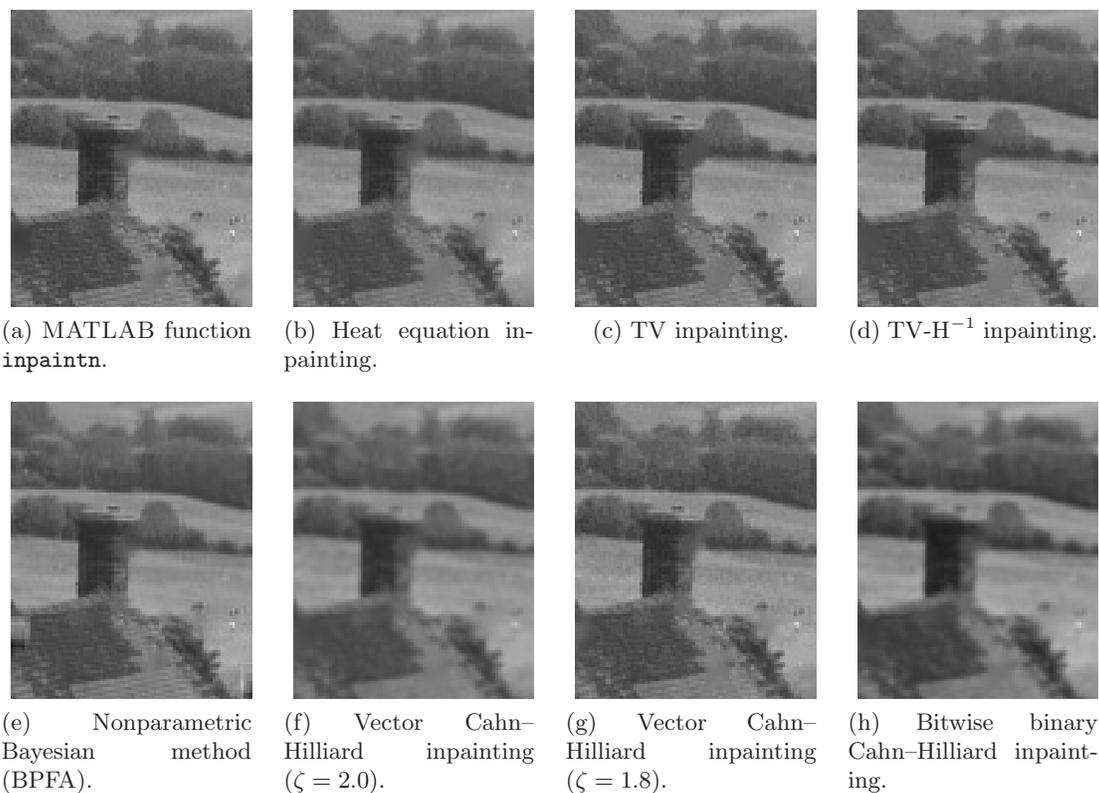


Figure 6. *Inpainted hill image using different inpainting models.*

The last test example is a 512×512 hill image;⁸ see Figure 5(a). Figure 5(b) displays an extract of the image after k-means clustering with 15 gray values. It zooms in the chimney

⁸The image is taken from <http://people.duke.edu/~mz31/Results/BPFAImage/>.

of the third white house from the left. This portion is taken for visual comparisons of the different inpainting methods. The damaged version of the whole image is illustrated in Figure 5(c). Figure 6 illustrates the inpainted (extracted) images using the different approaches mentioned above. The right column of Table 3 lists the total number of iterations, the total computational times (in seconds), as well as the PSNR and SSIM values. In all Cahn–Hilliard approaches, we set $\omega_0 = 10^9$, $h_x = h_y = \frac{1}{512}$ and apply the ε -two-step procedure with a switch from $\varepsilon = 1$ to $\varepsilon = h_x$. The fractional Cahn–Hilliard model is performed with a fractional power of $\zeta = 1.8$. We run the heat equation–based approach with $\omega_0 = 10$ and both TV models with $\omega_0 = 10^3$, $\varepsilon = \frac{1}{512}$. We apply to all three methods the stopping criterion (6.1) with $\epsilon = 2 \cdot 10^{-5}$. We stop the MATLAB function `inpaintn` after $6 \cdot 10^4$ iterations. Again, TV and TV-H⁻¹ did not converge to the tolerance $\epsilon = 2 \cdot 10^{-5}$ within $6 \cdot 10^4$ iterations.

In summary, all examples confirm that the vector-valued Cahn–Hilliard inpainting model competes with existing inpainting methods. Moreover, the fractional Cahn–Hilliard inpainting approach turns out to be superior to its nonfractional version. It sharpens the image contours and improves the PSNR and SSIM values. To be more precise, by fixing all parameters and just varying the fractional power, such an improvement occurs if the fractional power is not too small. We observe that the interface might no longer be resolved fine enough for too small values of ζ . Spurious artifacts have occurred in the case $\zeta = 1.1$. In the next section, we further study the influence of the fractional power.

6.2. Influence of the fractional power. In this section, we further examine the effect of the fractional power with respect to the PSNR and SSIM values as well as the computational time. We experiment with the gray value stripe example from the last section but this time with a larger ε in the first run of the ε -two-step approach and a larger fidelity parameter ω_0 . We fix $\tau = 1$, $\omega_0 = 10^6$, $h_x = h_y = \frac{1}{64}$ and apply the ε -two-step procedure with a switch from $\varepsilon = 10$ to $\varepsilon = h_x$. We consider the fractional powers 2.0, 1.9, 1.7, 1.5, 1.3, 1.1. Figure 7 shows extracted portions of the inpainted images obtained by Cahn–Hilliard inpainting with the different fractional powers. Table 4 contains the total number of iterations, the total computational times (in seconds), the average computational times per time step (in seconds), the PSNR and SSIM values, the minimum and maximum pixel values of the final image, as well as the numbers of grid points within the interface in $\Omega \setminus D$ (in D).

Decreasing the fractional power increases the sharpness of the image. This effect improves the inpainting result to a certain degree. Starting from the standard power 2.0 we reach the optimum at the fractional power 1.5. The PSNR and SSIM values as well as the visual image are at their highest level. A further decrease of the fractional power in turn worsens the results. One possible reason might be the stopping criterion (6.1), as already pointed out in the previous section. It seems that it aborts the iteration too early for small values of the fractional power. Hence, we will compare different stopping criteria in the next section.

Another explanation might be a too strong influence of the regularization regarding the fractional power ζ . Hence, it could prevent the successful connection of the stripes across the large gap. Moreover, the size of the interfacial parameter plays an important role. In the last section, we observed spurious artifacts when the interface was not resolved fine enough. Therefore, a possible solution might be an increasing interfacial parameter during a decreasing fractional power. Also, the influence of the fidelity parameter ω_0 should match the fractional

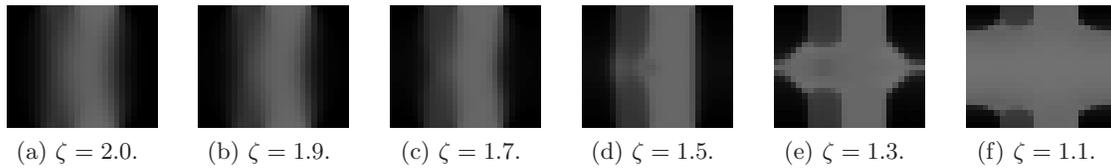


Figure 7. Inpainted gray value stripe image using different fractional powers ζ .

Table 4

Cahn–Hilliard inpainting results for the gray value stripe inpainting using different fractional powers ζ . The total number of iterations (*Iter.*), total computational times in seconds (*CPU*), average computational times per time step in seconds ($\overline{\text{CPU}}$), PSNR and SSIM values, minimum (*min*) and maximum (*max*) pixel values of the final image, as well as the numbers of grid points within the interface in $\Omega \setminus D$ (in *D*).

ζ	Iter.	CPU (s)	$\overline{\text{CPU}}$ (s)	PSNR	SSIM	min	max	Interface
2.0	2144	44.21	0.0206	22.92	0.7975	$-3.13 \cdot 10^{-2}$	1.0313	2 (8)
1.9	2576	52.71	0.0205	24.05	0.8336	$-3.06 \cdot 10^{-2}$	1.0307	2 (7)
1.7	3942	80.42	0.0204	26.70	0.8791	$-2.76 \cdot 10^{-2}$	1.0277	2 (6)
1.5	6287	128.46	0.0204	28.95	0.8999	$-5.83 \cdot 10^{-3}$	1.0059	0 (4)
1.3	9476	192.86	0.0204	21.21	0.8174	$-7.38 \cdot 10^{-4}$	1.0007	0 (-)
1.1	6186	126.51	0.0205	15.43	0.7069	$-7.63 \cdot 10^{-5}$	1.0001	0 (-)

power. Decreasing fractional powers strengthens a second sharpness effect besides the one caused by the fidelity parameter. This results in a slowdown of the convergence. Hence, a decreasing fidelity parameter during a decreasing fractional power should fix a too strong sharpening effect. We discuss the issue of the parameter influences in section 6.4. A resulting topic for future discussion might be a varying fractional power throughout the simulation similar to the ε -two-step approach.

6.3. Stopping criteria. This section compares different stopping criteria. We experiment with the gray value stripe image from the previous section. We choose the same parameters as before and focus on Cahn–Hilliard inpainting with a fractional power of 1.3. Besides the criterion in (6.1) we test five other stopping criteria:

$$(6.3) \quad \left(\sqrt{\sum_{i=1}^N \|\vec{U}_i^{(n)} - \vec{U}_i^{(n-1)}\|_2^2} \right)^{\zeta/2} / \left(\sqrt{\sum_{i=1}^N \|\vec{U}_i^{(n-1)}\|_2^2} \right)^{\zeta/2} \leq \epsilon,$$

$$(6.4) \quad \sqrt{\sum_{i=1}^N \|\text{vec}(\Lambda^{\zeta/2} * (U_i^{(n)} - U_i^{(n-1)})^2)\|_1} / \sqrt{\sum_{i=1}^N \|\text{vec}(\Lambda^{\zeta/2} * (U_i^{(n-1)})^2)\|_1} \leq \epsilon,$$

$$(6.5) \quad \sqrt{\sum_{i=1}^N \|\vec{U}_i^{(n)} - \vec{U}_i^{(n-1)}\|_2^2} \leq \epsilon,$$

$$(6.6) \quad \left(\sqrt{\sum_{i=1}^N \|\vec{U}_i^{(n)} - \vec{U}_i^{(n-1)}\|_2^2} \right)^{\zeta/2} \leq \epsilon,$$

$$(6.7) \quad \sqrt[\zeta]{\sum_{i=1}^N \|\vec{U}_i^{(n)} - \vec{U}_i^{(n-1)}\|_\zeta^\zeta} \leq \epsilon.$$

Note we use the notation $\|\mathbf{U}\|_2 = \sqrt{\sum_{i=1}^N \|U_i\|_2^2}$ for the 2-norm of the phases U_i in matrix form (as presented in section 5). Then $\|U_i\|_2$ equals the largest singular value of U_i . We denote by $\Lambda^{\zeta/2} = (\lambda_{\alpha,\beta}^{\zeta/2}) \in \mathbb{R}^{m_y \times m_x}$ the matrix containing the eigenvalues of the fractional Laplacian. In (6.4), we write \cdot^* and \cdot^2 for the pointwise multiplication and squaring. Further, $\|\vec{U}_i\|_p$ denotes the p -norm of the phases U_i in vector form.

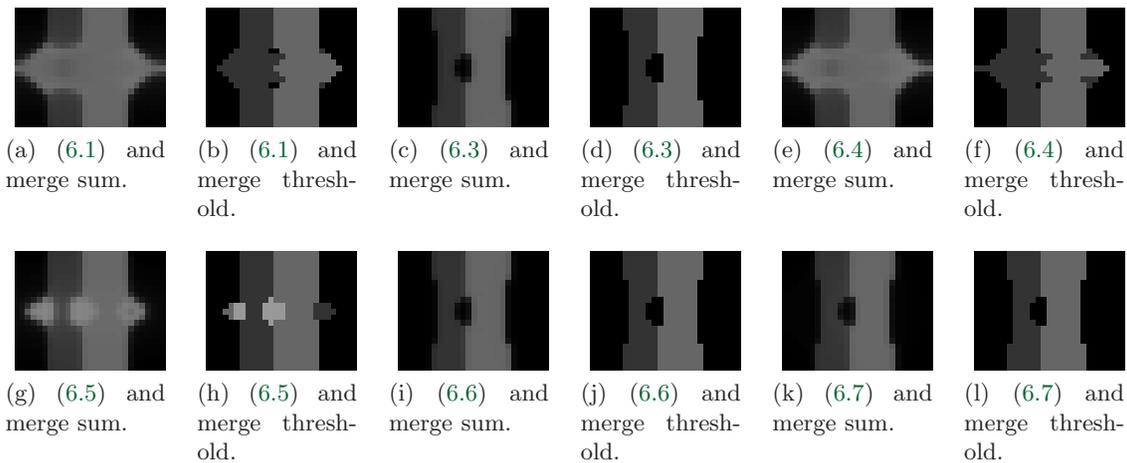


Figure 8. Inpainted gray value stripe image using different stopping and merging criteria.

Besides different stopping criteria, we also compare two different merging techniques to obtain the final image. Our proposed approach (merge sum) is given in (6.2). The second approach is based on thresholding (merge thresholding) and used in [25]: It replaces $U_K^{(n)}(\alpha, \beta) = \max_{i=1, \dots, N} U_i^{(n)}(\alpha, \beta)$ by 1 and $U_j^{(n)}(\alpha, \beta)$ by 0 for all $j \neq K$. Figure 8 shows extracted portions of the inpainted images obtained by Cahn–Hilliard inpainting using the different stopping and merging criteria. Table 5 lists the stopping tolerances ϵ_1, ϵ_2 for the first and second runs (due to the ϵ -two-step approach), the total number of iterations, the total computational times (in seconds), as well as the PSNR and SSIM values for the two merging techniques.

We observe that none of the stopping criteria could produce a satisfying result. The threshold merging technique results in almost every case in slightly better PSNR/SSIM values.

Table 5

Cahn–Hilliard inpainting results for the gray value stripe inpainting using different stopping and merging criteria. The stopping tolerances ϵ_1, ϵ_2 , total number of iterations (*Iter.*), total computational times in seconds (*CPU*), average computational times per time step in seconds ($\overline{\text{CPU}}$), as well as the PSNR and SSIM values for two merging techniques.

Stopping criterion						Merge sum		Merge threshold	
	ϵ_1	ϵ_2	Iter.	CPU (s)	$\overline{\text{CPU}}$ (s)	PSNR	SSIM	PSNR	SSIM
(6.1)	$2 \cdot 10^{-4}$	$2 \cdot 10^{-5}$	9476	192.51	0.0204	21.21	0.8174	24.80	0.9066
(6.3)	$2 \cdot 10^{-4}$	$2 \cdot 10^{-5}$	54857	936.05	0.0171	30.81	0.9490	30.24	0.9455
(6.4)	$2 \cdot 10^{-4}$	$2 \cdot 10^{-5}$	10500	178.77	0.0170	21.32	0.8228	23.93	0.9033
(6.5)	$1 \cdot 10^{-2}$	$1 \cdot 10^{-3}$	17570	280.04	0.0159	24.90	0.8759	26.34	0.9327
(6.6)	$1 \cdot 10^{-2}$	$1 \cdot 10^{-3}$	28020	450.21	0.0161	31.45	0.9543	30.91	0.9537
(6.7)	$1 \cdot 10^{-2}$	$1 \cdot 10^{-3}$	20032	387.01	0.0193	31.69	0.9465	30.91	0.9537

However, this is only due to the fact that it removes the interface. Visually, this approach is also not able to merge the stripe ends across the damaged region in the expected natural way. Our observations suggest that the parameter choice plays a crucial role. As already noticed in section 6.2, the parameter setting should be matched to the fractional power. This will be discussed in the next section.

6.4. Influence of the fractional power, the interfacial parameter, and the fidelity parameter. In the following, we study the influence of the fractional power ζ , the interfacial parameter ε , and the fidelity parameter ω_0 . We experiment with the gray value stripe image from section 6.2. This time we will not use the same parameter values for different fractional powers. Instead, we try to adapt ε and ω_0 to the fractional setting. As seen in sections 6.2 and 6.1, the smaller the fractional power, the smaller the interfacial width. We have observed spurious artifacts when the interface was not resolved fine enough. Hence, we aim to increase ε while decreasing the fractional power. In this experiment, we will only vary ε in the second run of the ε -two-step approach. For the first run, we will fix $\varepsilon = 0.1$ for all fractional powers. Varying the parameter ε in both runs is a topic of future research.

Further, the influence of the fidelity parameter ω_0 should match the fractional power. Decreasing fractional powers strengthens a second sharpness effect besides the one caused by the fidelity parameter. Hence, we aim to decrease the fidelity parameter while decreasing the fractional power.

Table 6 lists the parameter settings, total number of iterations, the total computational times (in seconds), the average computational times per time step (in seconds), the PSNR and SSIM values, as well as the minimum and maximum pixel values of the final image. We have numerically worked out the optimal parameter setting for each fractional power. In all cases, the final image shows a perfect straight conjunction of all stripe ends. Note that there might be better choices of parameter settings in the sense of larger PSNR/SSIM values. However, the table confirms our assumption regarding the influence between ζ , ε , ω_0 . Further, note that the interface size in all examples is zero. We suggest, that the chosen parameter settings provide a limit case of optimal outputs. A slight change of ε or ω_0 can lead to spurious effects. Hence, reaching a zero interface width is always an intricate experiment. In practice, one would

Table 6

Cahn–Hilliard inpainting results for the gray value stripe inpainting using different parameters. The fractional powers ζ , interfacial parameters ε , fidelity parameters ω_0 , total number of iterations (Iter.), total computational times in seconds (CPU), average computational times per time step in seconds ($\overline{\text{CPU}}$), PSNR and SSIM values, as well as minimum (min) and maximum (max) pixel values of the final image.

ζ	ε	ω_0	Iter.	CPU (s)	$\overline{\text{CPU}}$ (s)	PSNR	SSIM	min	max
2.0	$0.04 h_x$	$2 \cdot 10^5$	640	13.13	0.0205	58.97	0.9999	$-1.14 \cdot 10^{-3}$	1.0011
1.9	$0.07 h_x$	$1 \cdot 10^5$	627	13.07	0.0208	53.75	0.9998	$-1.81 \cdot 10^{-3}$	1.0018
1.7	$0.09 h_x$	$5 \cdot 10^4$	720	14.72	0.0204	58.07	0.9999	$-8.32 \cdot 10^{-4}$	1.0008
1.5	$0.30 h_x$	$2 \cdot 10^4$	871	17.35	0.0199	46.38	0.9991	$-2.27 \cdot 10^{-3}$	1.0023
1.3	$0.20 h_x$	$9 \cdot 10^3$	1025	21.16	0.0206	61.69	1.0000	$-1.16 \cdot 10^{-4}$	1.0001
1.1	$1.00 h_x$	$5 \cdot 10^3$	1610	33.06	0.0205	44.21	0.9962	$-1.54 \cdot 10^{-3}$	1.0016

start with the nonfractional ($\zeta = 2$) model and choose the parameters such that at least a few grid points lie within the interface. The interfacial profile of the original Cahn–Hilliard equation ($\omega_0 \equiv 0$) can be described by means of a tanh term: Following [48], the authors define the interface thickness to be the distance from 0.05 to 0.95, so that the equilibrium interface thickness is given by $\frac{4\sqrt{2}\varepsilon}{\tanh(0.9)}$. If we want to have at least eight grid points across the interface, we get the condition $\varepsilon \approx h$, where h denotes the mesh size. Regarding the initial value of ε (i.e., the first run of the ε -two-step approach), the authors of [7] propose to choose it nearly equal to the numerical maximum gap spacing. The value of ω_0 depends on, among others, ε . For small values of ε , ω_0 must be chosen very large to guarantee continuity of edges; see [7] for a detailed discussion on this topic. Having such a parameter setting for the nonfractional model, one can reduce the fractional power up to a certain degree such that the PSNR/SSIM values are improved.

6.5. Demonstration of the Gibbs simplex condition. This section demonstrates numerically the fulfillment of the Gibbs simplex condition. We consider the gray value stripe examples from section 6.2. For each of them, we sum up the final phase variables and note the minimum and maximum values. These are shown in Table 7 together with the minimum and maximum value among all single final phase variables.

Table 7

Cahn–Hilliard inpainting results for the gray value stripe inpainting using different fractional powers; see section 6.2. The minimum (Sum min) and maximum (Sum max) values of the sum of all final phase variables, as well as minimum (Phase min) and maximum (Phase max) values among all single final phase variables.

Example	Sum min	Sum max	Phase min	Phase max
Figure 7(a)	0.99999999999181	1.00000000000000	-0.0315	1.0462
Figure 7(b)	0.999999999998986	1.00000000000001	-0.0307	1.0308
Figure 7(c)	0.999999999998522	1.00000000000000	-0.0277	1.0279
Figure 7(d)	0.999999999997705	1.00000000000000	-0.0059	1.0059
Figure 7(e)	0.999999999996501	1.00000000000000	-0.0023	1.0008
Figure 7(f)	0.999999999997809	1.00000000000000	-0.0001	1.0001

6.6. Metal artifact reduction. One major challenge in X-ray CT is metal artifact reduction; see, e.g., [43]. Metallic objects implanted in the human body cause severe streak artifacts in CT images. These in turn significantly deteriorate the image quality. Projection interpolation methods [49, 23] correct metal artifacts directly in sinogram space. The main steps of those algorithms are as follows:

1. Given is the original projection data (sinogram).
2. Reconstruct the CT image using filtered back projection (FBP).
3. Segment the metal regions.
4. Reproject the metal regions to determine the missing data in the sinogram.
5. Correct the missing projection data.
6. Reconstruct the image from the inpainted sinogram using the FBP.
7. Insert the previously segmented metallic objects.

Our aim is to apply Cahn–Hilliard inpainting in step 5. The following experiment uses a variant of the Shepp–Logan (S-L) phantom. Five metal regions with much higher attenuation were added into the S-L phantom (256×256); see Figure 9(a). The parameters for the modified phantom are taken from [68, p. 5]. Figure 9(b) shows the sinogram of the phantom. The reconstructed phantom image using the FBP suffers from streak artifacts caused by the metal objects; see Figure 9(c). Using a simple thresholding method we can extract the metal regions. Note that because this paper focuses on the inpainting problem we do not study this segmentation problem. Next, we locate the metal regions in the projection data set (sinogram). These parts form the missing data. We employ Cahn–Hilliard inpainting to create an inpainted sinogram. Figures 9(d), 9(e), and 9(f) illustrate the inpainted sinograms for the standard and two fractional Cahn–Hilliard inpainting approaches. Finally, we reconstruct the phantom image from the inpainted sinogram using the FBP; see Figures 9(g), 9(h), and 9(i).

Compared to the reconstructed image in Figure 9(c), the metal artifacts were significantly reduced after inpainting. In this example, the passage from nonfractional to fractional inpainting does not significantly improve the image quality. Based on the experience of the former experiments, we suggest that a more suitable parameter setting could improve the results. However, finding the optimal parameters still remains a topic of future work. The PSNR value is 28.187 after standard Cahn–Hilliard inpainting, 28.199 after fractional Cahn–Hilliard inpainting with a fractional power of $\zeta = 1.8$, and 28.203 after fractional Cahn–Hilliard inpainting with a fractional power of $\zeta = 1.6$. The used model parameters are $\omega_0 = 10^8$, $h_x = h_y = \frac{1}{367}$. For the initialization, we have segmented the sinogram (367×180) into 40 gray values using k-means clustering. The ε -two-step procedure was applied with a switch from $\varepsilon = 1000$ to $\varepsilon = 10h_x$. Using the stopping criterion (6.1), the standard model converged after 869 iterations and a CPU time of 3280.47s, the fractional model with $\zeta = 1.8$ after 2008 iterations and 7445.06s, and the fractional model with $\zeta = 1.6$ after 3205 iterations and 12124.30s.

6.7. Three-dimensional visualization of medical images. The use of Fourier spectral methods allows a straightforward extension to higher spatial dimensions. As an example, we consider the three-dimensional visualization of MRI images; see [57, 64, 38, 11]. Parts of the human body are given in the form of a sequence of slices. Often, the distance between these slices is significantly larger than the image pixel size. Hence, it is necessary to interpolate additional slices in order to obtain an accurate three-dimensional description. This in turn

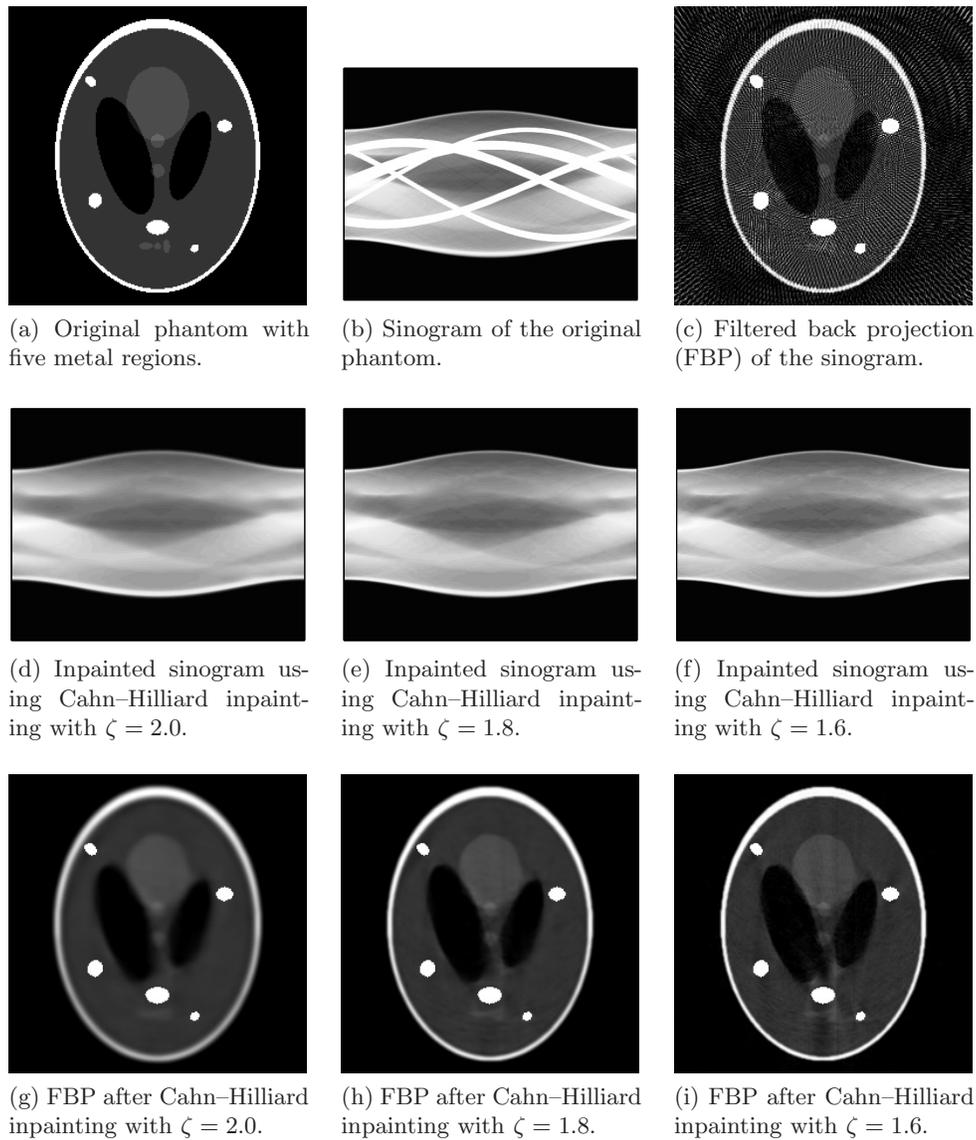


Figure 9. Metal artifact reduction using Cahn–Hilliard inpainting.

helps medical professionals make more accurate diagnoses.

The following example uses the MRI data set that comes with MATLAB. It comprises a scan of a human cranium ($128 \times 128 \times 27$) in the form of 27 horizontal slices of size 128×128 each. For this experiment, we work with the first ten slices, which are illustrated in Figure 10.

Our aim is to create four virtual slices between every two slices. This results in a total number of 46 horizontal slices. The inpainting method acts now in a three-dimensional cube. Ten horizontal planes provide us with image information. The remaining 26 horizontal planes form the damaged region. We work with 20 gray values such that the resulting problem size is

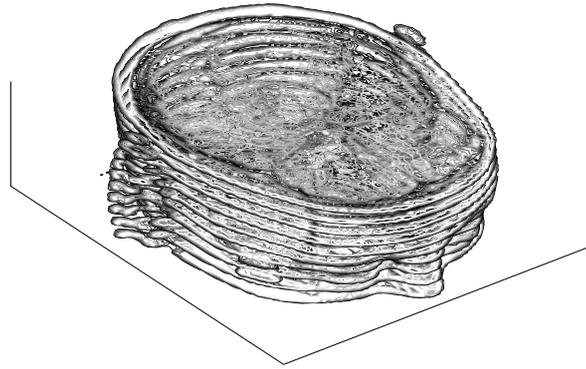
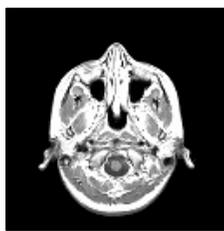
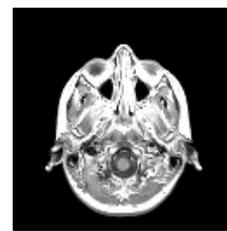


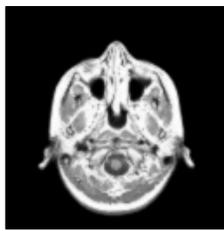
Figure 10. Ten magnetic resonance imaging slices of a human cranium.



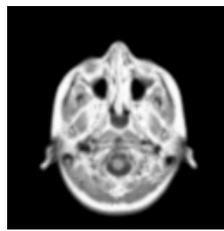
(a) Original slice 1.



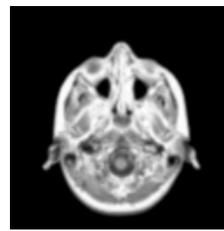
(b) Original slice 2.



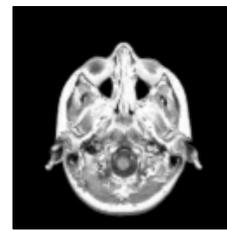
(c) Virtual slice 1.2.



(d) Virtual slice 1.4.



(e) Virtual slice 1.6.



(f) Virtual slice 1.8.

Figure 11. Construction of virtual magnetic resonance imaging slices using Cahn–Hilliard inpainting.

$20 \times 128 \times 128 \times 46$. Figures 11(a) and 11(b) show two consecutive slices from the original MRI data set. Between them, four virtual slices were reconstructed using fractional Cahn–Hilliard inpainting; see Figures 11(c)–11(f). We set $\omega_0 = 10^9$, $h_x = h_y = \frac{1}{128}$, $h_z = \frac{2.5h_x}{4}$, $\zeta = 1.8$ and apply the ε -two-step procedure with a switch from $\varepsilon = 1000$ to $\varepsilon = h_x$.

Using the stopping criterion (6.1), Cahn–Hilliard inpainting converged after 1100 iterations and a CPU time of 16840.13s. The PSNR and SSIM value regarding the first slice are 36.98 and 0.9971.

7. Conclusions. In this paper, we have developed an inpainting model based on the vector-valued Cahn–Hilliard equation. This approach generalizes Bertozzi, Esedoğlu, and Gillette’s [8] binary Cahn–Hilliard inpainting model to gray value images. In addition, we have fur-

ther generalized our approach to a fractional-in-space version. This is done by replacing the standard differential operator by its fractional counterpart. We have adopted the definition of the fractional Laplace operator via the spectral decomposition of the Neumann Laplace operator, as motivated by [14]. Fourier spectral methods provide efficient solvers since they yield a fully diagonal scheme. Furthermore, their application to three spatial dimensions is straightforward.

The numerical results have shown the superiority of the fractional Cahn–Hilliard approach over its nonfractional version. This generalized version can be treated as a regularization of the standard model. It is characterized by a sharpness effect and improves the image quality. In particular, we observe an increase of the peak signal-to-noise ratio and structural similarity index. Likewise, the experiments confirm that the proposed model is competitive with previous inpainting methods, such as the total variation inpainting approach and its fourth-order variant.

Acknowledgment. The authors would like to thank Luise Blank as well as the anonymous referees for their helpful comments and suggestions.

REFERENCES

- [1] E. E. ADAMS AND L. W. GELHAR, *Field study of dispersion in a heterogeneous aquifer: 2. Spatial moments analysis*, Water Resour. Res., 28 (1992), pp. 3293–3307.
- [2] S. M. ALLEN AND J. W. CAHN, *A microscopic theory for antiphase boundary motion and its application to antiphase domain coarsening*, Acta Metall., 27 (1979), pp. 1085–1095.
- [3] W. BAATZ, M. FORNASIER, P. A. MARKOWICH, AND C.-B. SCHÖNLIEB, *Inpainting of ancient Austrian frescoes*, in Bridges Leeuwarden: Mathematics, Music, Art, Architecture, Culture, R. Sarhangi and C. H. Séquin, eds., London, 2008, Tarquin Publications, pp. 163–170.
- [4] J. BAI AND X.-C. FENG, *Fractional-order anisotropic diffusion for image denoising*, IEEE Trans. Image Process., 16 (2007), pp. 2492–2502.
- [5] D. A. BENSON, S. W. WHEATCRAFT, AND M. M. MEERSCHAERT, *Application of a fractional advection-dispersion equation*, Water Resour. Res., 36 (2000), pp. 1403–1412.
- [6] M. BERTALMIÓ, G. SAPIRO, V. CASELLES, AND C. BALLESTER, *Image inpainting*, in Proceedings of the ACM Computer Graphics and Interactive Techniques, SIGGRAPH '00, ACM Press, Addison–Wesley, New York, 2000, pp. 417–424.
- [7] A. BERTOZZI, S. ESEDOĞLU, AND A. GILLETTE, *Analysis of a two-scale Cahn–Hilliard model for binary image inpainting*, Multiscale Model. Simul., 6 (2007), pp. 913–936.
- [8] A. L. BERTOZZI, S. ESEDOĞLU, AND A. GILLETTE, *Inpainting of binary images using the Cahn–Hilliard equation*, IEEE Trans. Image Process., 16 (2007), pp. 285–291.
- [9] A. L. BERTOZZI AND A. FLENNER, *Diffuse interface models on graphs for classification of high dimensional data*, Multiscale Model. Simul., 10 (2012), pp. 1090–1118.
- [10] L. BLANK, M. BUTZ, AND H. GARCKE, *Solving the Cahn–Hilliard variational inequality with a semi-smooth Newton method*, ESAIM Control Optim. Calc. Var., 17 (2011), pp. 931–954.
- [11] A. G. BORS, L. KECHAGIAS, AND I. PITAS, *Binary morphological shape-based interpolation applied to 3-D tooth reconstruction*, IEEE Trans. Med. Imag., 21 (2002), pp. 100–108.
- [12] J. BOSCH AND M. STOLL, *Preconditioning for vector-valued Cahn–Hilliard equations*, SIAM J. Sci. Comput., 37 (2015), to appear.
- [13] J. BOSCH, D. KAY, M. STOLL, AND A. J. WATHEN, *Fast solvers for Cahn–Hilliard inpainting*, SIAM J. Imaging Sci., 7 (2014), pp. 67–97.
- [14] A. BUENO-OROVIO, D. KAY, AND K. BURRAGE, *Fourier spectral methods for fractional-in-space reaction-diffusion equations*, BIT, 54 (2014), pp. 937–954.

- [15] A. BUENO-OROVIO, D. KAY, V. GRAU, B. RODRIGUEZ, AND K. BURRAGE, *Fractional diffusion models of cardiac electrical propagation: Role of structural heterogeneity in dispersion of repolarization*, J. R. Soc. Interface, 11 (2014), 20140352.
- [16] M. BURGER, L. HE, AND C.-B. SCHÖNLIEB, *Cahn–Hilliard inpainting and a generalization for grayvalue images*, SIAM J. Imaging Sci., 2 (2009), pp. 1129–1167.
- [17] J. W. CAHN AND J. E. HILLIARD, *Free energy of a nonuniform system. I. Interfacial free energy*, J. Chem. Phys., 28 (1958), pp. 258–267.
- [18] 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.
- [19] T. CHAN AND J. SHEN, *Variational restoration of nonflat image features: Models and algorithms*, SIAM J. Appl. Math., 61 (2001), pp. 1338–1361.
- [20] T. F. CHAN AND J. SHEN, *Nontexture inpainting by curvature-driven diffusions*, J. Vis. Commun. Image Represent., 12 (2001), pp. 436–449.
- [21] T. F. CHAN AND J. SHEN, *Mathematical models for local nontexture inpaintings*, SIAM J. Appl. Math., 62 (2002), pp. 1019–1043.
- [22] T. F. CHAN AND J. SHEN, *Image Processing and Analysis: Variational, PDE, Wavelet, and Stochastic Methods*, SIAM, Philadelphia, 2005.
- [23] Y. CHEN, YI. LI, H. GUO, Y. HU, L. LUO, X. YIN, J. GU, AND C. TOUMOULIN, *CT metal artifact reduction method based on improved image segmentation and sinogram in-painting*, Math. Probl. Eng., 2012 (2012), 786281.
- [24] Y. CHEN AND X. YE, *Projection onto a Simplex*, arXiv preprint, arXiv:1101.6081, 2011.
- [25] L. CHERFILS, H. FAKIH, AND A. MIRANVILLE, *A Cahn–Hilliard system with a fidelity term for color image inpainting*, J. Math. Imaging Vision, (2015), pp. 1–15.
- [26] L. CHERFILS, H. FAKIH, AND A. MIRANVILLE, *On the Bertozzi–Esedoglu–Gillette–Cahn–Hilliard equation with logarithmic nonlinear terms*, SIAM J. Imaging Sci., 8 (2015), pp. 1123–1140.
- [27] M. I. M. COPETTI AND C. M. ELLIOTT, *Numerical analysis of the Cahn–Hilliard equation with a logarithmic free energy*, Numer. Math., 63 (1992), pp. 39–65.
- [28] J. A. DOBROSOTSKAYA AND A. L. BERTOZZI, *A wavelet-Laplace variational technique for image deconvolution and inpainting*, IEEE Trans. Image Process., 17 (2008), pp. 657–663.
- [29] C. M. ELLIOTT, *The Cahn–Hilliard model for the kinetics of phase separation*, in Mathematical Models for Phase Change Problems, J. F. Rodrigues, ed., Internat. Ser. Numer. Math. 88, Birkhäuser, Basel, 1989, pp. 35–73.
- [30] C. M. ELLIOTT AND S. LUCKHAUS, *A Generalised Diffusion Equation for Phase Separation of a Multi-component Mixture with Interfacial Free Energy*, preprint 195, University of Bonn, 1991.
- [31] C. M. ELLIOTT AND A. M. STUART, *The global dynamics of discrete semilinear parabolic equations*, SIAM J. Numer. Anal., 30 (1993), pp. 1622–1663.
- [32] S. ESEDOĞLU AND J. SHEN, *Digital inpainting based on the Mumford–Shah–Euler image model*, European J. Appl. Math., 13 (2002), pp. 353–370.
- [33] D. J. EYRE, *Systems of Cahn–Hilliard equations*, SIAM J. Appl. Math., 53 (1993), pp. 1686–1712.
- [34] D. J. EYRE, *An Unconditionally Stable One-Step Scheme for Gradient Systems*, Technical report, Department of Mathematics, University of Utah, Salt Lake City, UT, 1998.
- [35] D. GARCIA, *Robust smoothing of gridded data in one and higher dimensions with missing values*, Comput. Statist. Data Anal., 54 (2010), pp. 1167–1178.
- [36] C. GARCIA-CARDONA, E. MERKURJEV, A. L. BERTOZZI, A. FLENNER, AND A. G. PERCUS, *Multiclass data segmentation using diffuse interface methods on graphs*, IEEE Trans. Pattern Anal. Mach. Intell., 36 (2014), pp. 1600–1613.
- [37] H. GARCKE, B. NESTLER, AND B. STOTH, *On anisotropic order parameter models for multi-phase systems and their sharp interface limits*, Phys. D, 115 (1998), pp. 87–108.
- [38] A. GOSHTASBY, D. A. TURNER, AND L. V. ACKERMAN, *Matching of tomographic slices for interpolation*, IEEE Trans. Med. Imag., 11 (1992), pp. 507–516.
- [39] J. GU, L. ZHANG, G. YU, Y. XING, AND Z. CHEN, *X-ray CT metal artifacts reduction through curvature based sinogram inpainting*, J. X-ray Sci. Technol., 14 (2006), pp. 73–82.
- [40] J. E. HILLIARD AND J. W. CAHN, *An evaluation of procedures in quantitative metallography for volume-fraction analysis*, Trans. Am. Inst. Min. Metall. Eng., 221 (1961), pp. 344–352.

- [41] M. E. HOCHSTENBACH AND L. REICHEL, *Fractional Tikhonov regularization for linear discrete ill-posed problems*, BIT, 51 (2011), pp. 197–215.
- [42] M. ILIC, F. LIU, I. TURNER, AND V. ANH, *Numerical approximation of a fractional-in-space diffusion equation*, I, Fract. Calc. Appl. Anal., 8 (2005), pp. 323–341.
- [43] W. A. KALENDER, R. HEBEL, AND J. EBERSBERGER, *Reduction of CT artifacts caused by metallic implants*, Radiology, 164 (1987), pp. 576–577.
- [44] E. KLANN AND R. RAMLAU, *Regularization by fractional filter methods and data smoothing*, Inverse Problems, 24 (2008), 025018.
- [45] K. KRUSE AND A. IOMIN, *Superdiffusion of morphogens by receptor-mediated transport*, New J. Phys., 10 (2008), 023019.
- [46] S. LARNIER AND R. MECCA, *Fractional-order diffusion for image reconstruction*, in IEEE International Conference on Acoustics, Speech and Signal Processing, 2012, pp. 1057–1060.
- [47] H. G. LEE, J.-W. CHOI, AND J. KIM, *A practically unconditionally gradient stable scheme for the N-component Cahn–Hilliard system*, Phys. A, 391 (2012), pp. 1009–1019.
- [48] H. G. LEE AND J. KIM, *A second-order accurate non-linear difference scheme for the N-component Cahn–Hilliard system*, Phys. A, 387 (2008), pp. 4787–4799.
- [49] R. M. LEWITT AND R. H. T. BATES, *Image reconstruction from projections. III. Projection completion methods (theory)*, Optik, 50 (1978), pp. 189–204.
- [50] Y. LI, D. JEONG, J.-I. CHOI, S. LEE, AND J. KIM, *Fast local image inpainting based on the Allen–Cahn model*, Digit. Signal Process., 37 (2015), pp. 65–74.
- [51] E. MERKURJEV, C. GARCIA-CARDONA, A. L. BERTOZZI, A. FLENNER, AND A. G. PERCUS, *Diffuse interface methods for multiclass segmentation of high-dimensional data*, Appl. Math. Lett., 33 (2014), pp. 29–34.
- [52] E. MERKURJEV, T. KOSTIĆ, AND A. L. BERTOZZI, *An MBO scheme on graphs for classification and image processing*, SIAM J. Imaging Sci., 6 (2013), pp. 1903–1930.
- [53] R. METZLER AND J. KLAFTER, *The random walk’s guide to anomalous diffusion: A fractional dynamics approach*, Phys. Rep., 339 (2000), pp. 1–77.
- [54] R. H. NOCHETTO, E. OTÁROLA, AND A. J. SALGADO, *A PDE approach to fractional diffusion in general domains: A priori error analysis*, Found. Comput. Math., 15 (2015), pp. 733–791.
- [55] K. B. OLDHAM AND J. SPANIER, *The Fractional Calculus: Theory and Applications of Differentiation and Integration to Arbitrary Order*, Math. Sci. Engrg. 111, Elsevier, Burlington, MA, 1974.
- [56] Y. OONO AND S. PURI, *Study of phase-separation dynamics by use of cell dynamical systems. I. Modeling*, Phys. Rev. A, 38 (1988), pp. 434–453.
- [57] S. R. PIRES, E. L. FLORES, C. A. Z. BARCELOS, AND M. A. BATISTA, *Interpolation of computerized tomography slices using 3D digital inpainting*, in 19th Brazilian Symposium on Computer Graphics and Image Processing, Manaus, Amazonas, IEEE, 2006, pp. 129–138.
- [58] S. SALSA, *Partial Differential Equations in Action: From Modelling to Theory*, Universitext, Springer, Mailand, 2009.
- [59] S. G. SAMKO, A. A. KILBAS, AND O. I. MARICHEV, *Fractional Integrals and Derivatives: Theory and Applications*, Gordon and Breach, Yverdon, 1993.
- [60] L. SARBU, *Primal-Dual Active Set Methods for Allen–Cahn Variational Inequalities*, Ph.D. thesis, University of Sussex, UK, 2010.
- [61] C.-B. SCHÖNLIEB AND A. BERTOZZI, *Unconditionally stable schemes for higher order inpainting*, Commun. Math. Sci., 9 (2011), pp. 413–457.
- [62] A. TSAI, A. YEZZI, JR., AND A. S. WILLSKY, *Curve evolution implementation of the Mumford–Shah functional for image segmentation, denoising, interpolation, and magnification*, IEEE Trans. Image Process., 10 (2001), pp. 1169–1186.
- [63] G. WANG, D. GARCIA, Y. LIU, R. DE JEU, AND A. J. DOLMAN, *A three-dimensional gap filling method for large geophysical datasets: Application to global satellite soil moisture observations*, Environ. Model. Softw., 30 (2012), pp. 139–142.
- [64] Y. WANG, Z. ZHANG, AND B. GUO, *3D image interpolation based on directional coherence*, in IEEE Workshop on Mathematical Methods in Biomedical Image Analysis, Kauai, HI, IEEE, 2001, pp. 195–202.
- [65] Z. WANG, A. C. BOVIK, H. R. SHEIKH, AND E. P. SIMONCELLI, *Image quality assessment: From error*

- visibility to structural similarity*, IEEE Trans. Image Process., 13 (2004), pp. 600–612.
- [66] Q. YANG, F. LIU, AND I. TURNER, *Numerical methods for fractional partial differential equations with Riesz space fractional derivatives*, Appl. Math. Model., 34 (2010), pp. 200–218.
- [67] P. YI-FEI, *Application of fractional differential approach to digital image processing*, J. Sichuan Univ. Eng. Sci. Ed., 3 (2007), 022.
- [68] Y. ZHANG, Y.-F. PU, J.-R. HU, Y. LIU, Q.-L. CHEN, AND J.-L. ZHOU, *Efficient CT metal artifact reduction based on fractional-order curvature diffusion*, Comput. Math. Methods Med., 2011 (2011), 173748.
- [69] Y. ZHANG, Y.-F. PU, J.-R. HU, AND J.-L. ZHOU, *A class of fractional-order variational image inpainting models*, Appl. Math. Inf. Sci., 6 (2012), pp. 299–306.
- [70] M. ZHOU, H. CHEN, J. PAISLEY, L. REN, L. LI, Z. XING, D. DUNSON, G. SAPIRO, AND L. CARIN, *Nonparametric Bayesian dictionary learning for analysis of noisy and incomplete images*, IEEE Trans. Image Process., 21 (2012), pp. 130–144.