Max Planck Institute Magdeburg
Preprints

Jessica Bosch Martin Stoll

A fractional inpainting model based on the vector-valued Cahn–Hilliard equation

MPIMD/15-04 **March 26, 2015**

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 fractional approach over the classical one. 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.

Imprint:

Max Planck Institute for Dynamics of Complex Technical Systems, Magdeburg

Publisher:
Max Planck Institute for
Dynamics of Complex Technical Systems

Address:
Max Planck Institute for
Dynamics of Complex Technical Systems
Sandtorstr. 1
39106 Magdeburg

http://www.mpi-magdeburg.mpg.de/preprints/

1. Introduction

Three-dimensional visualizations of medical images help the professionals to make more accurate diagnoses. Parts of the human body are given in 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 [37, 7, 24].

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 [2], the replacement of selected objects or the reduction of artifacts in medical images [25]. 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. [4] introduced the image inpainting technique into digital image processing. Their third-order nonlinear PDE propagates the 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, among others the total variation (TV) inpainting model [14, 12], inpainting based on curvature-driven diffusions [13], Euler's elastica inpainting model [41], the Mumford–Shah inpainting model [43, 14] or the Mumford–Shah–Euler inpainting model [20]. We refer the reader to [15] for an overview of existing inpainting methods. In summary, Schönlieb et al. [40] 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 [48, 3]. 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 [36]. They are used to describe many real world phenomena, e.g. the anomalous diffusion process [34] or dispersion in heterogeneous aquifers [1]. 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) [3, 50, 31, 49]. 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 [30, 27], which are applied among other things in denoising.

1.3. Basic model

Bertozzi et al. [6, 5] introduced the fourth-order Cahn–Hilliard inpainting approach for binary images. They take 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 et al's [40] bitwise binary inpainting and Burger et al's [10] 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. Secondly, 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. In contrast to bitwise binary inpainting, the gray values interact with each other. Hence, the reconstructed images stay in the gray value range of the original image. As a further improvement, we generalize our model to a fractional-in-space version, called fractional Cahn–Hilliard inpainting. Numerical examples emphasize the superiority of the fractional approach over the classical one. 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 et al's [6, 5] 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 at al. [40] pointed out that the idea of convexity splitting [19, 22] 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. Now in Section 5, we exploit the spectral decomposition of the occurring (fractional) Laplace operator as proposed by Bueno-Orovio et al. [9]. Hence, we can take advantage of the FFT similar to the binary inpainting approach [6]. 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 derive the convexity conditions for the application of the convexity splitting method in Appendix A. Finally, in Appendix B we prove the unconditional stability of our time-discrete scheme.

2. Binary Cahn–Hilliard inpainting

We start with reflecting on Bertozzi et al's [6, 5] black-and-white Cahn–Hilliard inpainting model. Let f be the given binary image which is defined on the image domain $\Omega \subset \mathbb{R}^d$ $(d = 2, 3)$. 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 $u\colon \Omega \times (0, T) \to \mathbb{R}$ of f is constructed by following the evolution of

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

$$\nabla u \cdot \mathbf{n} = \nabla(\Delta u) \cdot \mathbf{n} = 0 \quad \text{on } \partial\Omega, \tag{2}$$

where

$$\omega = \omega(\mathbf{x}) = \begin{cases} 0 & \text{if } \mathbf{x} \in D, \\ \omega_0 & \text{if } \mathbf{x} \in \Omega \setminus D. \end{cases} \tag{3}$$

Here, $T > 0$ is a fixed time. The choice $\omega \equiv 0$ leads to the original Cahn–Hilliard equation, which is a well-known model for coarsening and phase separation processes [26, 11]. In this case, the variable u describes the evolution 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. For proper modeling, ε has to be chosen proportional to the mesh size of the space discretization. The evolution of the interface is driven by an interfacial energy whose minimization penalizes interfaces with high curvatures. The term $\varepsilon\Delta u$ in (1) describes the part coming from the interfacial energy.

The potential function ψ in (1) 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. Typical examples of potential functions are given in the following paragraph. Equation (2) imposes the natural zero Neumann boundary condition as well as the mass conserving boundary condition. Here, $\mathbf{n}$ is the unit normal vector to $\partial\Omega$ pointing outwards from Ω. Next, we want to use the Cahn–Hilliard model's characteristics for the inpainting problem.

Regarding image processing, the two phases A and B represent the colors black and white. Their interface consists of gray values and forms a smooth transition. Now, let us consider the case $\omega \not\equiv 0$. Bertozzi et al. [6, 5] introduced (1)-(2) as the modified Cahn–Hilliard equation. As can be seen from (3), the added fidelity term $\omega(f - u)$ only acts in the regions outside the inpainting domain. Hence, it keeps the solution u close to the given image f in the undamaged parts. 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 minimizing the curvature of their interface at the same time. Finally, the inpainted image is constructed by following this evolution to steady state. Well-known potential functions are the smooth double-well potential [6]

$$\psi(u) = u^2(u-1)^2, \tag{4}$$

and the nonsmooth double obstacle potential [8]

$$\psi(u) = \begin{cases} \frac{1}{2}u(1-u) & \text{if } 0 \le u \le 1, \\ \infty & \text{otherwise.} \end{cases} \tag{5}$$

Recently, logarithmic potentials have been considered [17]. 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 [21, 18]. Let f be a given gray value image defined on Ω, N the number of desired gray values and $\mathbf{g} = (g_1, \dots, g_N)^\top$ the vector of desired gray values. Note that $2 \le N \le 256$. A vector-valued phase variable $\mathbf{u} = (u_1, \dots, u_N)^\top : \Omega \times (0,T)$ is introduced. The component u_i describes the evolution of gray value g_i for $i = 1, \dots, N$. This means, 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

$$\sum_{i=1}^{N} u_i = 1 \tag{6}$$

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

$$\mathcal{G}^N := \left\{ \mathbf{v} \in \mathbb{R}^N \;\middle|\; \sum_{i=1}^{N} v_i = 1,\ v_i \ge 0 \text{ for } i = 1, \dots, N \right\}. \tag{7}$$

As the Cahn–Hilliard equation is a phase field model, the evolution of $\mathbf{u}$ develops a smooth gray value transition. That means, the region between gray values has a certain width, the interface, as already mentioned above in the binary Cahn–Hilliard model. We propose the following generalization of (1)–(2) for gray value images

$$\partial_t u_i = -\Delta\left(\varepsilon \Delta u_i - \frac{1}{\varepsilon}\frac{\partial \psi}{\partial u_i}(\mathbf{u}) - \frac{1}{\varepsilon}\beta(\mathbf{u})\right) + \omega(f_i - u_i), \tag{8}$$

$$\nabla u_i \cdot \mathbf{n} = \nabla(\Delta u_i) \cdot \mathbf{n} = 0 \quad \text{on } \partial\Omega, \tag{9}$$

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

of gray value g_i in f for $i = 1, \ldots, N$. Comparing (8)–(9) with (1)–(2), it can be seen that the generalization from the scalar to the vector-valued model is more than just a straightforward extension. In fact, the presence of the term $\beta(\mathbf{u})$, which is defined as

$$\beta(\mathbf{u}) := -\frac{1}{N} \sum_{i=1}^{N} \frac{\partial \psi}{\partial u_i}(\mathbf{u}),$$

is needed to ensure (6) in the inpainting domain D.

Remark Note that (8)–(9) 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 $L = L(\mathbf{u}) = (L_{ij}(\mathbf{u}))_{i,j=1,\ldots,N}$ is required in the corresponding model. This is for example the case, if the mobility in the interface is larger than in the pure phases. Nevertheless, Lee et al. [32] 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.

This works concentrates on a smooth potential function and the well-known potential function (4) generalizes to

$$\psi(\mathbf{u}) = \frac{1}{4} \sum_{i=1}^{N} u_i^2 (1 - u_i)^2. \tag{10}$$

4. Convexity splitting

In the case of black-and-white Cahn–Hilliard inpainting, Bertozzi et al. [6] 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 et al. [40] proved consistency, unconditional stability and convergence of this scheme. The convexity splitting method was originally introduced by Elliott and Stuart [19] and is often attributed to Eyre [22]. 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 (1)–(2) and (8)–(9), as described further on.

In the following, we extend the numerical analysis of the convexity splitting scheme for the scalar inpainting model (1)–(2) studied in [40] to the vector-valued inpainting model (8)–(9). As mentioned above, the modified Cahn–Hilliard equation is not given by a gradient flow. The original multicomponent Cahn–Hilliard equation, i.e. (8)–(9) with $\omega \equiv 0$, is the H^{-1}-gradient flow for the Ginzburg–Landau energy

$$\mathcal{E}_1(\mathbf{u}) = \int_\Omega \left\{ \frac{\varepsilon}{2} \sum_{i=1}^{N} |\nabla u_i|^2 + \frac{1}{\varepsilon} \psi(\mathbf{u}) \right\} \mathrm{d}\mathbf{x}, \tag{11}$$

while the fidelity term is derived from the L^2-gradient flow for the energy

$$\mathcal{E}_2(\mathbf{u}) = \int_\Omega \left\{ \frac{\omega}{2} \sum_{i=1}^{N} (f_i - u_i)^2 \right\} \mathrm{d}\mathbf{x}. \tag{12}$$

The original idea of convexity splitting applied to gradient systems is to write the considered energy functional as sum of a convex plus a concave energy functional. The convex part is then treated implicitly whilst the concave part is treated explicitly. Under the right conditions, this approach leads to an unconditionally gradient stable time-discretization scheme.

Regarding the evolution we deal with, i.e. a sum of two gradients of two different energies $\mathcal{E}_1$ and $\mathcal{E}_2$, 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

$$\mathcal{E}_{1c}(\mathbf{u}) = \int_\Omega \left\{ \frac{\varepsilon}{2} \sum_{i=1}^{N} |\nabla u_i|^2 + \frac{C_1}{2} \sum_{i=1}^{N} u_i^2 \right\} \mathrm{d}\mathbf{x},$$

$$\mathcal{E}_{1e}(\mathbf{u}) = \int_\Omega \left\{ -\frac{1}{\varepsilon}\psi(\mathbf{u}) + \frac{C_1}{2} \sum_{i=1}^{N} u_i^2 \right\} \mathrm{d}\mathbf{x},$$

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

$$\mathcal{E}_{2c}(\mathbf{u}) = \int_\Omega \left\{ \frac{C_2}{2} \sum_{i=1}^{N} u_i^2 \right\} \mathrm{d}\mathbf{x},$$

$$\mathcal{E}_{2e}(\mathbf{u}) = \int_\Omega \left\{ -\frac{\omega}{2} \sum_{i=1}^{N} (f_i - u_i)^2 + \frac{C_2}{2} \sum_{i=1}^{N} u_i^2 \right\} \mathrm{d}\mathbf{x}.$$

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, 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. Even, the proof is similar and we refer the reader to Appendix A.

The resulting discrete time-stepping scheme is given by

$$\frac{\mathbf{u}^{(n)} - \mathbf{u}^{(n-1)}}{\tau} = -\nabla_{H^{-1}}\Big(\mathcal{E}_{1c}(\mathbf{u}^{(n)}) - \mathcal{E}_{1e}(\mathbf{u}^{(n-1)})\Big) - \nabla_{L^2}\Big(\mathcal{E}_{2c}(\mathbf{u}^{(n)}) - \mathcal{E}_{2e}(\mathbf{u}^{(n-1)})\Big),$$

where $\nabla_{H^{-1}}$ and ∇_{L^2} represent the gradient descents with respect to the H^{-1}- and L^2-inner product, respectively. 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

$$\begin{aligned}
&\frac{u_i^{(n)} - u_i^{(n-1)}}{\tau} + \varepsilon\Delta^2 u_i^{(n)} - C_1\Delta u_i^{(n)} + C_2 u_i^{(n)} \\
&= \frac{1}{\varepsilon}\Delta\left(\frac{\partial\psi}{\partial u_i}(\mathbf{u}^{(n-1)})\right) - \frac{1}{\varepsilon N}\Delta\left(\sum_{j=1}^{N}\frac{\partial\psi}{\partial u_j}(\mathbf{u}^{(n-1)})\right) + \omega(f_i - u_i^{(n-1)}) - C_1\Delta u_i^{(n-1)} + C_2 u_i^{(n-1)},
\end{aligned} \tag{13}$$

for $i = 1, \ldots, N$. For the proof of the unconditional stability of the scheme (13), we refer the reader to Appendix B.

5. Fourier spectral methods

Bertozzi et al. [6] took benefit 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 will not make use of the finite differences but still of the FFT. The idea comes from the work of Bueno-Orovio et al. [9] 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 (13) to a more general form by replacing the standard Laplace operator by its fractional counterpart

$$\begin{aligned}\frac{u_i^{(n)} - u_i^{(n-1)}}{\tau} &+ \varepsilon \Delta^{\zeta} u_i^{(n)} + C_1 (-\Delta)^{\frac{\zeta}{2}} u_i^{(n)} + C_2 u_i^{(n)} \\ &= -\frac{1}{\varepsilon}(-\Delta)^{\frac{\zeta}{2}} \left(\frac{\partial \psi}{\partial u_i}(\mathbf{u}^{(n-1)}) \right) + \frac{1}{\varepsilon N}(-\Delta)^{\frac{\zeta}{2}} \left(\sum_{j=1}^{N} \frac{\partial \psi}{\partial u_j}(\mathbf{u}^{(n-1)}) \right) \\ &\quad + \omega(f_i - u_i^{(n-1)}) + C_1 (-\Delta)^{\frac{\zeta}{2}} u_i^{(n-1)} + C_2 u_i^{(n-1)}. \quad (14)\end{aligned}$$

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

As pointed out in [35, 47], 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 [39]. 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 [28]. As already mentioned above, this is the interpretation we adopt. Let us now summarize the main elements of the spectral approach according to [9] for the two-dimensional case. Three spatial dimensions can be constructed straightforward.

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 [38, Theorem 8.6]. More precisely, we consider the following eigenvalue problem in $\Omega = [0, a] \times [0, b]$

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

$\alpha, \beta = 1, 2, \ldots$ 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 [9], we define

$$\mathcal{U}_\zeta := \Bigg\{ u = \sum_{\alpha=1}^{\infty} \sum_{\beta=1}^{\infty} \hat{u}(\alpha,\beta)\varphi_{\alpha,\beta} \in L^2(\Omega),\ \hat{u}(\alpha,\beta) = (u, \varphi_{\alpha,\beta}),$$
$$\sum_{\alpha=1}^{\infty} \sum_{\beta=1}^{\infty} |\hat{u}(\alpha,\beta)|^2 |\lambda_{\alpha,\beta}|^{\frac{\zeta}{2}} < \infty,\ 0 < \zeta \leq 2 \Bigg\}, \quad (15)$$

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

$$(-\Delta)^{\frac{\zeta}{2}} u = \sum_{\alpha=1}^{\infty} \sum_{\beta=1}^{\infty} \hat{u}(\alpha,\beta) \lambda_{\alpha,\beta}^{\frac{\zeta}{2}} \varphi_{\alpha,\beta}, \quad (16)$$

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.

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-direction, hence not including boundary nodes. The mesh points are given by

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

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 $u_i^{(n)}$, $i = 1, \ldots N$, in (13) 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 $k = 1, \ldots, m_y$, $l = 1, \ldots, m_x$. We denote by $\hat{U}_i^{(n)}$ the two-dimensional 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 $i = 1, \ldots, N$. In the same manner of capital letters, we define the discrete versions of f_i. Applying the Fourier transform to both sides of (13) and using the spectral decomposition of the Laplacian (16), one gets

$$\hat{U}_i^{(n)}(\alpha,\beta) = \frac{\lambda_{\alpha,\beta}^{\frac{\zeta}{2}} \left[\left(-\frac{1}{\varepsilon} \frac{\partial \psi}{\partial u_i}(\mathbf{U}^{(n-1)}) + \frac{1}{\varepsilon N} \sum_{j=1}^{N} \frac{\partial \psi}{\partial u_j}(\mathbf{U}^{(n-1)}) \right)\hat{}\,(\alpha,\beta) + C_1 \hat{U}_i^{(n-1)}(\alpha,\beta) \right]}{\frac{1}{\tau} + \varepsilon \lambda_{\alpha,\beta}^{\zeta} + C_1 \lambda_{\alpha,\beta}^{\frac{\zeta}{2}} + C_2} + \frac{\left[\omega F_i + (C_2 - \omega) U_i^{(n-1)} \right]\hat{}\,(\alpha,\beta) + \frac{1}{\tau} \hat{U}_i^{(n-1)}(\alpha,\beta)}{\frac{1}{\tau} + \varepsilon \lambda_{\alpha,\beta}^{\zeta} + C_1 \lambda_{\alpha,\beta}^{\frac{\zeta}{2}} + C_2}, \quad (17)$$

for $i = 1, \ldots, N$ and $\alpha = 1, \ldots, m_y$, $\beta = 1, \ldots, m_x$. The system (17) is fully diagonal and the two-dimensional DCT matrices $\hat{U}_i^{(n)}$, $i = 1, \ldots, N$, 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 are the same for the standard and fractional Cahn–Hilliard inpainting model.

6. Numerical results

In this section, we present numerical results for the standard and fractional vector-valued Cahn–Hilliard inpainting model. For all simulations we use Bertozzi et al's [6] 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 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

$$\frac{\|\mathbf{U}^{(n)} - \mathbf{U}^{(n-1)}\|_2}{\|\mathbf{U}^{(n-1)}\|_2} \leq \epsilon, \quad (18)$$

if not mentioned otherwise. We choose $\epsilon = 2 \cdot 10^{-4}$ for the first step and $\epsilon = 2 \cdot 10^{-5}$ for the second one.

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 (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. Each phase variable represents one cluster. Secondly, we set every phase variable in the damaged regions to the value $\frac{1}{N}$. This assignment fulfills the conditions of the Gibbs simplex (7).

We use the PSNR

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

as well as the SSIM [46] to measure the quality of reconstruction[1]. Here, u_{orig} denotes the original image without damaged regions and u_{inpaint} the inpainted image. 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.

We provide the test results in high quality as well as the MATLAB® code to reproduce them as supplementary material with this paper.[2]

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 called `inpaintn`[3] [23, 44]. Note, the results obtained with bitwise binary inpainting do not lie in the image range $[0, 255]$ anymore. Therefore, we cannot (reasonably) measure the PSNR and SSIM value of the obtained results.

Schönlieb provides three inpainting codes[4]. 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 [51] non-parametric Bayesian method, which they term the beta process factor analysis (BPFA)[5]. In all examples below, we run BPFA with the default values.

In contrast to our proposed vector-valued model, the methods we compare with 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 zero, instead of $\frac{1}{N}$.

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. Figure 2 illustrates the inpainted images using the different approaches mentioned above. The middle column of Table 1 contains 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 $\omega_0 = 10^5$, $h_x = h_y = \frac{1}{100}$ 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 a fractional power of $\zeta = 1.6$. We run the heat equation based approach with

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

[2] http://epubs.siam.org/doi/...

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

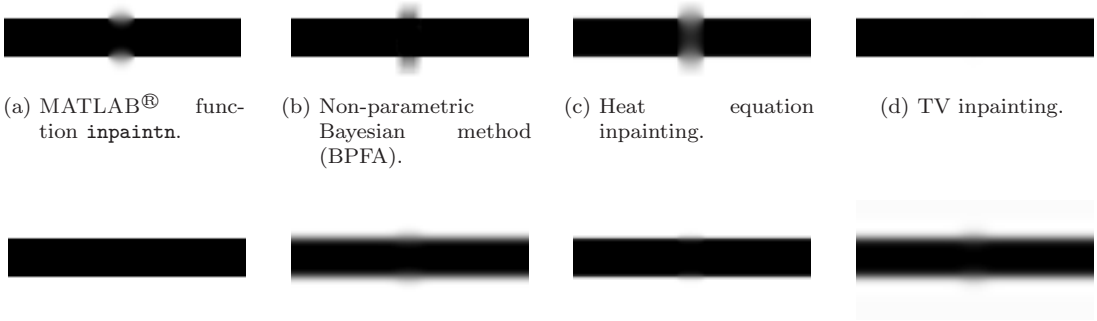
[4] http://www.mathworks.com/matlabcentral/fileexchange/34356-higher-order-total-variation-inpainting

[5] http://people.duke.edu/~mz31/Results/BPFAImage/

$\omega_0 = 10$ 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.

(a) Original image. (b) Damaged image. (c) Original image. (d) Damaged image.

Figure 1: Two simple test examples: A black-and-white stripe image and a gray value stripe image.



(a) MATLAB® function `inpaintn`. (b) Non-parametric Bayesian method (BPFA). (c) Heat equation inpainting. (d) TV inpainting.

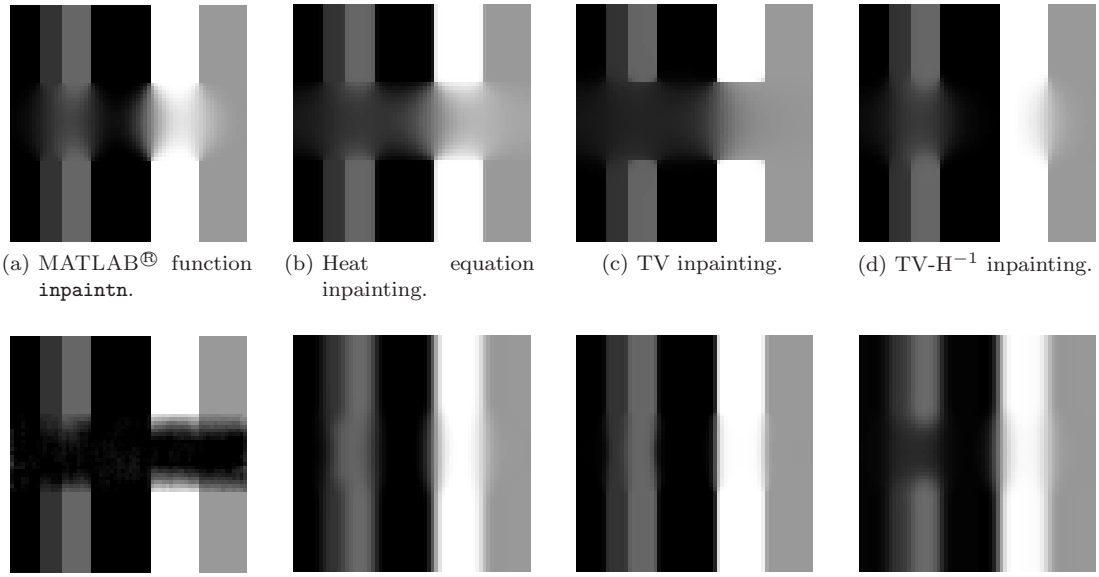
(e) TV-H^{-1} inpainting. (f) Standard Cahn–Hilliard inpainting. (g) Fractional Cahn–Hilliard inpainting ($\zeta = 1.6$). (h) Bitwise Cahn–Hilliard inpainting.

Figure 2: Inpainted black-and-white stripe image using different inpainting models.

	black-and-white stripe image				gray value stripe image			
inpainting method	iter	CPU (s)	PSNR	SSIM	iter	CPU (s)	PSNR	SSIM
`inpaintn`	100000	169.31	27.37	0.9480	100000	234.17	22.39	0.7839
BPFA	1072	246.91	26.67	0.9247	1072	143.48	12.16	0.6618
heat	1000000	110.91	23.65	0.9010	7248	6.42	18.89	0.7010
TV	100000	158.93	59.76	0.9975	27041	50.90	16.51	0.6427
TV-H^{-1}	100000	162.43	69.92	0.9999	85547	163.55	26.97	0.8268
vector CH ($\zeta = 2.0$)	100000	743.16	18.30	0.6220	1404	28.23	21.30	0.6846
vector CH ($\zeta = 1.6$)	100000	830.96	25.54	0.9499	1402	28.23	27.24	0.9192
bitwise CH	100000	3127.05	–	–	1112	34.81	–	–

Table 1: Results for the black-and-white and gray value stripe inpainting: The total number of iterations (iter), computational times in seconds (CPU) and PSNR/SSIM values using different inpainting models.

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, when we reduce the fractional power from 2 to 1.6. The image contours are sharpened and the PSNR/SSIM values are enhanced by a factor of about 1.5. However, regarding the computational time, the vector-valued approach is 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, 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.



(a) MATLAB® function `inpaintn`. (b) Heat equation inpainting. (c) TV inpainting. (d) TV-H^{-1} inpainting.

(e) Non-parametric Bayesian method (BPFA). (f) Cahn–Hilliard inpainting. (g) Fractional Cahn–Hilliard inpainting ($\zeta = 1.6$). (h) Bitwise binary Cahn–Hilliard inpainting.

Figure 3: Inpainted gray value stripe image using different inpainting models.

The second test example consists of six stripes spanning different widths and is of size 64×64, see Figure 1(c). It is composed of five gray values. Figure 1(d) shows the damaged version. Figure 3 illustrates the inpainted images using the different approaches mentioned above. The right column of Table 1 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 $\omega_0 = 10^5$, $h_x = h_y = \frac{1}{64}$ 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 a fractional power of $\zeta = 1.6$. 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 (18) 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 (18) might not be the optimal choice. In general, the discussion about the stopping criterion for the proposed Cahn–Hilliard approach should be a task for future work. Both, the standard and fractional Cahn–Hilliard inpainting model, provide a complete connection of the stripes over the inpainting domain. Again, the fractional approach beats the classical one. The PSNR/SSIM values are improved by a factor of 1.3.

The third test example is a 512×512 fingerprint image[6], see Figure 4(a). Figure 4(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 extract is taken for visual comparisons of the different inpainting methods. The damaged version of the whole image is illustrated in Figure 4(c). We have removed 80% of the pixels at random.

(a) Original image.

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

(c) Damaged image.

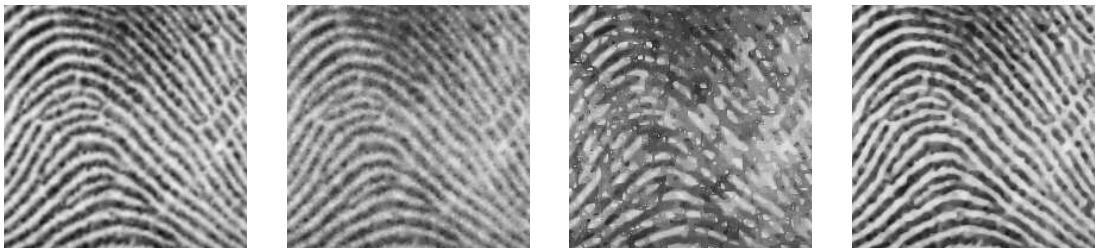
Figure 4: Fingerprint image.

Figure 5 illustrates the inpainted (extracted) images using the different approaches mentioned above. The middle column of Table 2 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

[6]The image is taken from http://people.duke.edu/~mz31/Results/BPFAImage/.

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 (18) with $\epsilon = 2 \cdot 10^{-5}$. We stop the MATLAB® function `inpaintn` after 10^4 iterations.

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 5(c) and 5(d) look unsatisfactory. If we would have continued the inpainting process the results would be of better quality, but at the expense of computational costs.



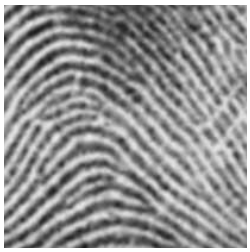
(a) MATLAB® function `inpaintn`.

(b) Heat equation inpainting.

(c) TV inpainting.

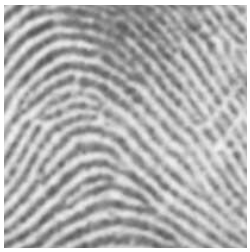
(d) TV-H^{-1} inpainting.

(e) Non-parametric Bayesian method (BPFA).



(f) Cahn–Hilliard inpainting.

(g) Fractional Cahn–Hilliard inpainting ($\zeta = 1.8$).



(h) Bitwise binary Cahn–Hilliard inpainting.

Figure 5: Inpainted fingerprint image using different inpainting models.

The last test example is a 512×512 hill image[7], see Figure 6(a). Figure 6(b) displays an extract of the image after k-means clustering with 15 gray values. It zooms in the chimney of third white house from the left. This extract is taken for visual comparisons of the different inpainting methods. The damaged version of the whole image is illustrated in Figure 6(c). Figure 7 illustrates the inpainted (extracted) images using the different approaches mentioned above. The right column of Table 2 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

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

with $\omega_0 = 10^3$, $\varepsilon = \frac{1}{512}$. We apply to these three methods the stopping criterion (18) 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.

	fingerprint image				hill image			
inpainting method	iter	CPU (s)	PSNR	SSIM	iter	CPU (s)	PSNR	SSIM
`inpaintn`	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	314	956.80	–	–	702	1982.41	–	–

Table 2: 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 $*$ marks 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 (18).

(a) Original image.

(b) Clustered extract of the original image using 15 gray values.

(c) Damaged image.

Figure 6: Hill image.

In summary, all examples confirm that the vector-valued Cahn–Hilliard inpainting model competes with existing inpainting methods. Moreover, the fractional approach turns out to be superior over the classical one. It sharpens the image contours and improves the PSNR and SSIM values. In the next section, we study the influence of the fractional power.

(a) MATLAB® function `inpaintn`.
(b) Heat equation inpainting.
(c) TV inpainting.
(d) TV-H^{-1} inpainting.

(e) Non-parametric Bayesian method (BPFA).
(f) Cahn–Hilliard inpainting.
(g) Fractional Cahn–Hilliard inpainting ($\zeta = 1.8$).
(h) Bitwise binary Cahn–Hilliard inpainting.

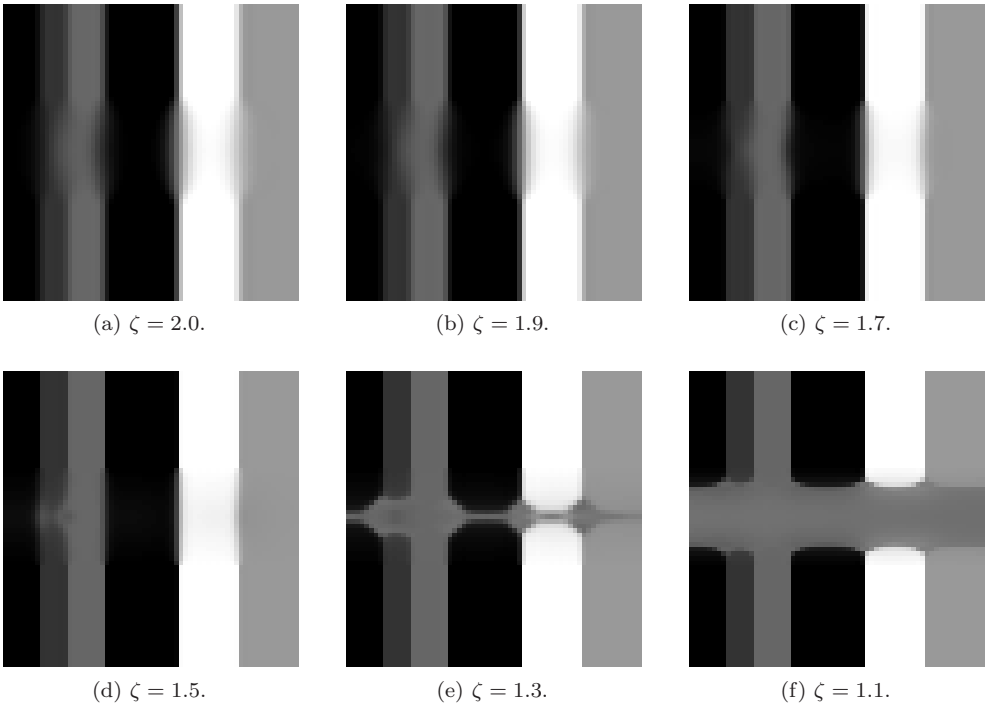
Figure 7: Inpainted hill image using different inpainting models.

6.2. Influence of the fractional power

In this section, we explore the effect of the fractional power with respect to the PSNR and SSIM value as well as the computational time. We experiment with the gray value stripe image from the last section, see Figure 1(c) and 1(d). We vary the fractional power and keep all the other model parameters fixed. 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 8 shows the inpainted images obtained by Cahn–Hilliard inpainting with the different fractional powers. Table 3 contains the total number of iterations, the total computational times (in seconds) as well as the PSNR and SSIM values.

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 value 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 (18), as already pointed out in the previous section. It seems that it aborts the iteration

too early for small values of the fractional power. 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. A resulting topic for future discussion might be a varying fractional power throughout the simulation similar to the ε-two-step approach.



(a) $\zeta = 2.0$. (b) $\zeta = 1.9$. (c) $\zeta = 1.7$.

(d) $\zeta = 1.5$. (e) $\zeta = 1.3$. (f) $\zeta = 1.1$.

Figure 8: Inpainted gray value stripe image using different fractional powers ζ.

ζ	iter	CPU (s)	PSNR	SSIM
2.0	2144	44.21	22.92	0.7975
1.9	2576	52.71	24.05	0.8336
1.7	3942	80.42	26.70	0.8791
1.5	6287	128.46	28.95	0.8999
1.3	9476	192.86	21.21	0.8174
1.1	6186	126.51	15.43	0.7069

Table 3: Cahn–Hilliard inpainting results for the gray value stripe inpainting: Total number of iterations (iter), computational times in seconds (CPU) and PSNR/SSIM values using different fractional powers ζ.

6.3. Metal artifact reduction

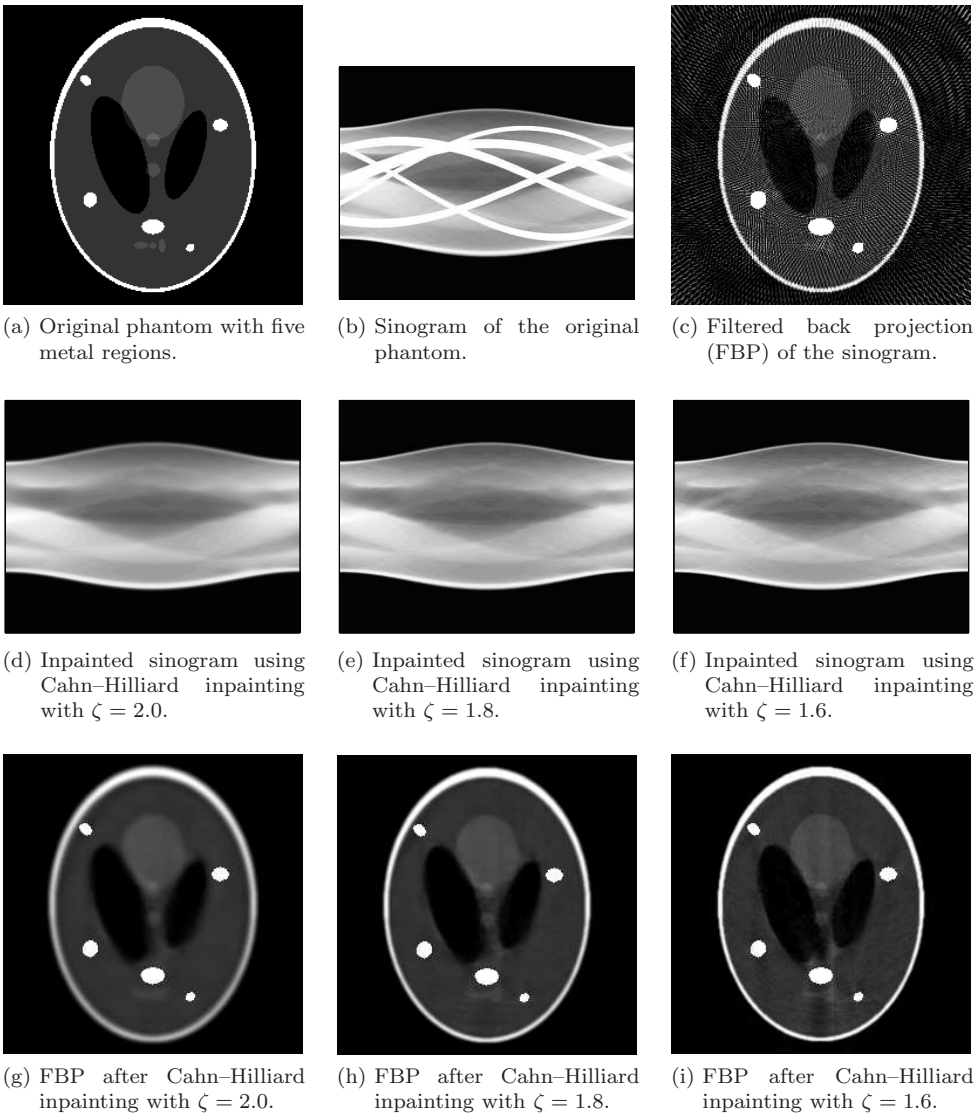
One major challenge in X-ray CT is the metal artifact reduction, see e.g. [29]. Metallic objects implanted in the human body cause severe streak artifacts in CT images. These in turn significantly deteriorate the image quality. Projection interpolation methods [33, 16] correct metal artifacts directly in sinogram space. The main steps of those algorithms are:

1. Given is the original projection data (sinogram).
2. Reconstruct the CT image using the 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 [49, 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, 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. Figure 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 Figure 9(g), 9(h) and 9(i).

Compared with the reconstructed image in Figure 9(c), the metal artifacts were significantly reduced after inpainting. Further, the passage from standard to fractional inpainting improves the image quality. 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 $\varepsilon = 10\, h_x$. Using the stopping criterion (18), 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 3295 iterations and 12124.30s.



(a) Original phantom with five metal regions.

(b) Sinogram of the original phantom.

(c) Filtered back projection (FBP) of the sinogram.

(d) Inpainted sinogram using Cahn–Hilliard inpainting with $\zeta = 2.0$.

(e) Inpainted sinogram using Cahn–Hilliard inpainting with $\zeta = 1.8$.

(f) Inpainted sinogram using Cahn–Hilliard inpainting with $\zeta = 1.6$.

(g) FBP after Cahn–Hilliard inpainting with $\zeta = 2.0$.

(h) FBP after Cahn–Hilliard inpainting with $\zeta = 1.8$.

(i) FBP after Cahn–Hilliard inpainting with $\zeta = 1.6$.

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

6.4. 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 also [37, 45, 24, 7]. Parts of the human body are given in 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 helps the professionals to 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 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.

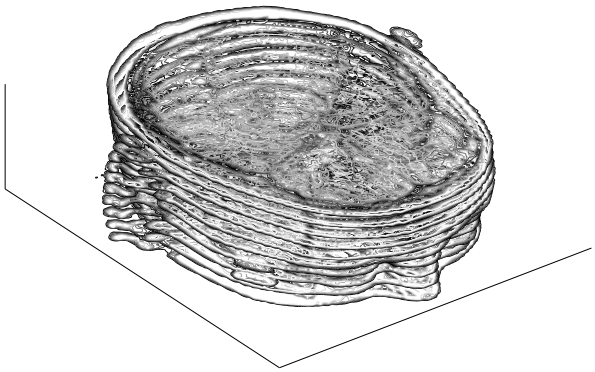
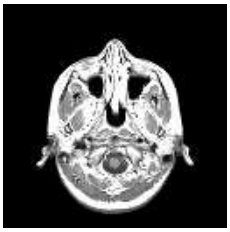


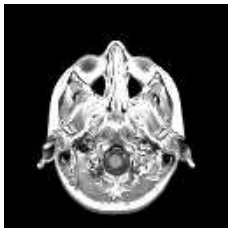
Figure 10: Ten magnetic resonance imaging slices of a human cranium.

Our aim is to create four virtual slices between each 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 $20 \times 128 \times 128 \times 46$. Figure 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 Figure 11(c)–11(f). We set $\omega_0 = 10^9$, $h_x = h_y = \frac{1}{128}$, $h_z = \frac{2.5\,h_x}{4}$, $\zeta = 1.8$ and apply the ε-two-step procedure with a switch from $\varepsilon = 1000$ to $\varepsilon = h_x$.

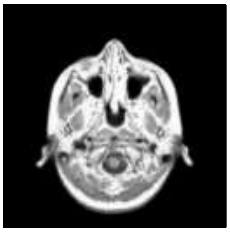
Using the stopping criterion (18), Cahn–Hilliard inpainting converged after 1100 iterations and a CPU time of 16840.13s. The PSNR and SSIM value regarding the first slice are 36.98 and 0.9971.



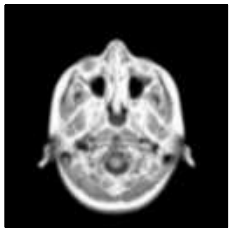
(a) Original slice 1.



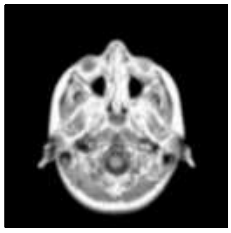
(b) Original slice 2.



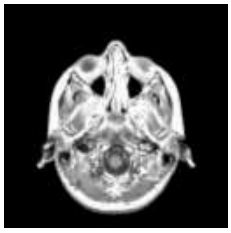
(c) Virtual slice 1.2.



(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.

7. Conclusions

In this paper, we have developed an inpainting model based on the vector-valued Cahn–Hilliard equation. This approach generalizes Bertozzi et al's [6] binary Cahn–Hilliard inpainting model to gray value images. Add to this, we have further 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 in [9]. 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 approach over the classical one. 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 competes with previous inpainting methods, such as the total variation inpainting approach and its fourth-order variant.

Acknowledgment

This research is supported by a research grant of the "International Max Planck Research School (IMPRS) for Advanced Methods in Process and System Engineering (Magdeburg)". The authors would like to thank Luise Blank for her helpful comments and suggestions.

A. Convexity splitting

We wish to prove here the requirements for the constants C_1 and C_2 to make sure that $\mathcal{E}_{1c}$, $\mathcal{E}_{1e}$, $\mathcal{E}_{2c}$ and $\mathcal{E}_{2e}$ are strictly convex. It is easy to see, that $\mathcal{E}_{1c}$ and $\mathcal{E}_{2c}$ are already strictly convex for $C_1 > 0$ and $C_2 > 0$. The crucial points are the energy functionals that contain the non-convex potential function $\psi(\mathbf{u})$ as well as the fidelity terms $\omega(f_i - u_i),\ i = 1, \dots, N$.

Lemma A.1 *Under the additional assumption that*

$$K = \max_{\mathbf{w} \in \mathbb{R}^N} \left| \frac{\partial^2 \psi}{\partial {w_i}^2}(\mathbf{w}) \right| \tag{19}$$

we have that $\mathcal{E}_{1e}$ and $\mathcal{E}_{2e}$ are strictly convex if C_1 is comparable to $\frac{1}{\varepsilon}$ and C_2 is comparable to ω_0.

Proof Based on [42, p. 54], we have to show

$$\mathcal{E}_{je}(\mathbf{u} + \mathbf{v}) - \mathcal{E}_{je}(\mathbf{u}) \geq \lim_{\delta \to 0} \frac{\mathcal{E}_{je}(\mathbf{u} + \delta \mathbf{v}) - \mathcal{E}_{je}(\mathbf{u})}{\delta}$$

for $j = 1, 2$. We have

$$\begin{aligned} \mathcal{E}_{1e}(\mathbf{u} + \mathbf{v}) - \mathcal{E}_{1e}(\mathbf{u}) &= \int_\Omega \left\{ -\frac{1}{\varepsilon} \left(\psi(\mathbf{u} + \mathbf{v}) - \psi(\mathbf{u}) \right) + \frac{C_1}{2} \sum_{i=1}^N \left((u_i + v_i)^2 - u_i^2 \right) \right\} \mathrm{d}\mathbf{x} \\ &= \int_\Omega \left\{ -\frac{1}{\varepsilon} \left(\psi(\mathbf{u} + \mathbf{v}) - \psi(\mathbf{u}) \right) + \frac{C_1}{2} \sum_{i=1}^N \left(v_i^2 + 2 u_i v_i \right) \right\} \mathrm{d}\mathbf{x}. \end{aligned} \tag{20}$$

As $\psi(\mathbf{u})$ is a smooth function, we can consider its Taylor expansion

$$\psi(\mathbf{u} + \mathbf{v}) = \psi(\mathbf{u}) + \sum_{i=1}^N \left(v_i \frac{\partial \psi}{\partial u_i}(\mathbf{u}) + \frac{1}{2} v_i^2 \frac{\partial^2 \psi}{\partial {w_i}^2}(\mathbf{w}) \right),$$

where $\mathbf{w}$ lies between $\mathbf{u} + \mathbf{v}$ and $\mathbf{u}$. Therefore, we obtain in (20)

$$\mathcal{E}_{1e}(\mathbf{u}+\mathbf{v}) - \mathcal{E}_{1e}(\mathbf{u}) = \int_\Omega \left\{ -\frac{1}{\varepsilon} \sum_{i=1}^N \left(v_i \frac{\partial \psi}{\partial u_i}(\mathbf{u}) + \frac{1}{2} v_i^2 \frac{\partial^2 \psi}{\partial {w_i}^2}(\mathbf{w}) \right) + \frac{C_1}{2} \sum_{i=1}^N \left(v_i^2 + 2 u_i v_i \right) \right\} \mathrm{d}\mathbf{x}.$$

Similarly, one can show

$$\lim_{\delta\to 0}\frac{\mathcal{E}_{1e}(\mathbf{u}+\delta\mathbf{v})-\mathcal{E}_{1e}(\mathbf{u})}{\delta}=\int_\Omega\left\{-\frac{1}{\varepsilon}\sum_{i=1}^N v_i\frac{\partial\psi}{\partial u_i}(\mathbf{u})+\frac{C_1}{2}\sum_{i=1}^N 2u_iv_i\right\}\mathrm{d}\mathbf{x},$$

which leads to

$$\begin{aligned}\mathcal{E}_{1e}(\mathbf{u}+\mathbf{v})-\mathcal{E}_{1e}(\mathbf{u})-\lim_{\delta\to 0}\frac{\mathcal{E}_{1e}(\mathbf{u}+\delta\mathbf{v})-\mathcal{E}_{1e}(\mathbf{u})}{\delta}&=\int_\Omega\left\{-\frac{1}{2\varepsilon}\sum_{i=1}^N v_i^2\frac{\partial^2\psi}{\partial {w_i}^2}(\mathbf{w})+\frac{C_1}{2}\sum_{i=1}^N v_i^2\right\}\mathrm{d}\mathbf{x}\\&\overset{(19)}{\geq}\int_\Omega\left\{\left(\frac{C_1}{2}-\frac{K}{2\varepsilon}\right)\sum_{i=1}^N v_i^2\right\}\mathrm{d}\mathbf{x}.\end{aligned}$$

Therefore, $\mathcal{E}_{1e}$ is strictly convex if C_1 is comparable to $\frac{1}{\varepsilon}$. Proceeding the same way with the second energy functional $\mathcal{E}_{2e}$ gives

$$\begin{aligned}\mathcal{E}_{2e}(\mathbf{u}+\mathbf{v})-\mathcal{E}_{2e}(\mathbf{u})-\lim_{\delta\to 0}\frac{\mathcal{E}_{2e}(\mathbf{u}+\delta\mathbf{v})-\mathcal{E}_{2e}(\mathbf{u})}{\delta}&=\int_\Omega\left\{\left(\frac{C_2}{2}-\frac{\omega}{2}\right)\sum_{i=1}^N v_i^2\right\}\mathrm{d}\mathbf{x}\\&\overset{\omega\leq\omega_0}{\geq}\int_\Omega\left\{\left(\frac{C_2}{2}-\frac{\omega_0}{2}\right)\sum_{i=1}^N v_i^2\right\}\mathrm{d}\mathbf{x}.\end{aligned}$$

Therefore, $\mathcal{E}_{2e}$ is strictly convex if $C_2>\omega_0$.

Remark Note, a similar assumption to (19) is made in the numerical analysis for the black-and-white Cahn–Hilliard inpainting model, see [40, Theorem 3.1]. In fact, the authors assume that the second derivative of the smooth potential in the previous time step is bounded.

B. Unconditional stability

We wish to prove here the unconditional stability of the scheme (13). In doing so, we follow [40] and extend the proof from the scalar to the vector-valued case. Throughout the following, we will make use of the $L^2(\Omega)$-inner product $(\cdot,\cdot)$ and of the scalar product

$$(\mathbf{u},\mathbf{v})=\int_\Omega\mathbf{u}\cdot\mathbf{v}\,\mathrm{d}\mathbf{x}=\sum_{i=1}^N(u_i,v_i)$$

in $L^2(\Omega)^N$. Additionally, we will write $\|\cdot\|$ for the norm in $L^2(\Omega)$ as well as in $L^2(\Omega)^N$ and use the notation $\nabla\mathbf{u}=(\nabla u_1,\dots,\nabla u_N)^\top$ and $\Delta\mathbf{u}=(\Delta u_1,\dots,\Delta u_N)^\top$.

Lemma B.1 *Let $\mathbf{u}^{(n)}$ be the n-th ($n\in\mathbb{N}$) iterate at time $n\tau$ for a time step $\tau>0$ of (13) with constants $C_1>\frac{1}{\varepsilon}$ and $C_2>\omega_0$. Additionally, we assume that*

$$\frac{\partial^2\psi}{\partial {u_i}^2}(\mathbf{u}^{(n-1)})\leq K \tag{21}$$

for a nonnegative constant K. Then the solution sequence $\mathbf{u}^{(n)}$ is bounded on a finite time interval $[0,T]$, for all $\tau > 0$. In particular for $n\tau \leq T$, $T > 0$ fixed, we have for every $\tau > 0$

$$\|\nabla \mathbf{u}^{(n)}\|^2 + \tau K_1 \|\Delta \mathbf{u}^{(n)}\|^2 \leq \mathrm{e}^{K_2 T} \left(\|\nabla \mathbf{u}^{(0)}\|^2 + \tau K_1 \|\Delta \mathbf{u}^{(0)}\|^2 + \tau C(\Omega, D, \omega_0, \mathbf{f}) \right), \tag{22}$$

for suitable constants K_1 and K_2, and constant C depending on Ω, D, ω_0, $\mathbf{f}$ only.

Proof By multiplying row i of the discrete model (13) with $-\Delta u_i^{(n)}$ and integrate over Ω, we obtain

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

Above, we have used Gauss's theorem together with the Neumann boundary conditions that are imposed in (9). In particular, Gauss's theorem applied to the integral containing the derivative of the potential function yields

$$\begin{aligned}
&\int_\Omega \Delta \left(\frac{\partial \psi}{\partial u_i}(\mathbf{u}^{(n-1)}) \right) \Delta u_i^{(n)} \mathrm{d}\mathbf{x} \\
&= \int_{\partial\Omega} \Delta u_i^{(n)} \nabla \left(\frac{\partial \psi}{\partial u_i}(\mathbf{u}^{(n-1)}) \right) \cdot \mathbf{n} \, \mathrm{d}\mathbf{s} - \int_\Omega \nabla \left(\frac{\partial \psi}{\partial u_i}(\mathbf{u}^{(n-1)}) \right) \cdot \nabla \Delta u_i^{(n)} \mathrm{d}\mathbf{x} \\
&= \int_{\partial\Omega} \Delta u_i^{(n)} \frac{\partial^2 \psi}{\partial {u_i}^2}(\mathbf{u}^{(n-1)}) \underbrace{\nabla u_i^{(n-1)} \cdot \mathbf{n}}_{=0} \mathrm{d}\mathbf{s} - \int_\Omega \frac{\partial^2 \psi}{\partial {u_i}^2}(\mathbf{u}^{(n-1)}) \nabla u_i^{(n-1)} \cdot \nabla \Delta u_i^{(n)} \mathrm{d}\mathbf{x} \\
&= -\left(\frac{\partial^2 \psi}{\partial u_i^2}(\mathbf{u}^{(n-1)}) \nabla u_i^{(n-1)}, \nabla \Delta u_i^{(n)} \right).
\end{aligned}$$

Using Young's inequality

$$\begin{aligned}
(\nabla u_i^{(n)}, \nabla u_i^{(n-1)}) &\leq \frac{1}{2}\|\nabla u_i^{(n)}\|^2 + \frac{1}{2}\|\nabla u_i^{(n-1)}\|^2, \\
\left(\frac{\partial^2 \psi}{\partial u_i^2}(\mathbf{u}^{(n-1)}) \nabla u_i^{(n-1)}, \nabla \Delta u_i^{(n)} \right) &\leq \frac{1}{2\delta_1} \left\| \frac{\partial^2 \psi}{\partial u_i^2}(\mathbf{u}^{(n-1)}) \nabla u_i^{(n-1)} \right\|^2 + \frac{\delta_1}{2} \left\| \nabla \Delta u_i^{(n)} \right\|^2, \\
\left(\Delta u_i^{(n-1)}, \Delta u_i^{(n)} \right) &\leq \frac{1}{2}\|\Delta u_i^{(n-1)}\|^2 + \frac{1}{2}\|\Delta u_i^{(n-1)}\|^2, \\
(\nabla \omega (f_i - u_i^{(n-1)}), \nabla u_i^{(n)}) &\leq \frac{1}{2}\|\nabla \omega (f_i - u_i^{(n-1)})\|^2 + \frac{1}{2}\|\nabla u_i^{(n)}\|^2,
\end{aligned}$$

we obtain

$$\left(\frac{1}{2\tau}+\frac{C_2}{2}-\frac{1}{2}\right)\|\nabla u_i^{(n)}\|^2+\frac{C_1}{2}\|\Delta u_i^{(n)}\|^2+\left(\varepsilon-\frac{\delta_1}{2\varepsilon}-\frac{\delta_2}{2\varepsilon}\right)\|\nabla\Delta u_i^{(n)}\|^2$$
$$\leq\left(\frac{1}{2\tau}+\frac{C_2}{2}\right)\|\nabla u_i^{(n-1)}\|^2+\frac{1}{2\delta_1\varepsilon}\left\|\frac{\partial^2\psi}{\partial u_i^2}(\mathbf{u}^{(n-1)})\nabla u_i^{(n-1)}\right\|^2$$
$$+\frac{1}{2\delta_2 N\varepsilon}\sum_{j=1}^{N}\left\|\frac{\partial^2\psi}{\partial u_j^2}(\mathbf{u}^{(n-1)})\nabla u_j^{(n-1)}\right\|^2+\frac{C_1}{2}\|\Delta u_i^{(n-1)}\|^2+\frac{1}{2}\|\nabla\omega(f_i-u_i^{(n-1)})\|^2,$$

where $\delta_1,\delta_2>0$. Using the estimate

$$\|\nabla\omega(f_i-u_i^{(n-1)})\|^2\leq 2\omega_0^2\|\nabla u_i^{(n-1)}\|^2+C(\Omega,D,\omega_0,f_i)$$

(see [40]) as well as $\delta_1=\delta_2=\varepsilon^2$, we obtain

$$\left(\frac{1}{2\tau}+\frac{C_2}{2}-\frac{1}{2}\right)\|\nabla u_i^{(n)}\|^2+\frac{C_1}{2}\|\Delta u_i^{(n)}\|^2$$
$$\leq\left(\frac{1}{2\tau}+\frac{C_2}{2}+\omega_0^2\right)\|\nabla u_i^{(n-1)}\|^2+\frac{1}{2\varepsilon^3}\left\|\frac{\partial^2\psi}{\partial u_i^2}(\mathbf{u}^{(n-1)})\nabla u_i^{(n-1)}\right\|^2$$
$$+\frac{1}{2N\varepsilon^3}\sum_{j=1}^{N}\left\|\frac{\partial^2\psi}{\partial u_j^2}(\mathbf{u}^{(n-1)})\nabla u_j^{(n-1)}\right\|^2+\frac{C_1}{2}\|\Delta u_i^{(n-1)}\|^2+C(\Omega,D,\omega_0,f_i).$$

Summing up these equations, we have

$$\left(\frac{1}{2\tau}+\frac{C_2}{2}-\frac{1}{2}\right)\|\nabla\mathbf{u}^{(n)}\|^2+\frac{C_1}{2}\|\Delta\mathbf{u}^{(n)}\|^2$$
$$\leq\left(\frac{1}{2\tau}+\frac{C_2}{2}+\omega_0^2\right)\|\nabla\mathbf{u}^{(n-1)}\|^2+\frac{1}{\varepsilon^3}\sum_{i=1}^{N}\left\|\frac{\partial^2\psi}{\partial u_i^2}(\mathbf{u}^{(n-1)})\nabla u_i^{(n-1)}\right\|^2$$
$$+\frac{C_1}{2}\|\Delta\mathbf{u}^{(n-1)}\|^2+C(\Omega,D,\omega_0,\mathbf{f}).$$

Because of Assumption (21) we can estimate

$$\left\|\frac{\partial^2\psi}{\partial u_i^2}(\mathbf{u}^{(n-1)})\nabla u_i^{(n-1)}\right\|^2\leq K^2\left\|\nabla u_i^{(n-1)}\right\|^2$$

(see [40]) and we have

$$\left(\frac{1}{2\tau}+\frac{C_2}{2}-\frac{1}{2}\right)\|\nabla\mathbf{u}^{(n)}\|^2+\frac{C_1}{2}\|\Delta\mathbf{u}^{(n)}\|^2$$
$$\leq\left(\frac{1}{2\tau}+\frac{C_2}{2}+\omega_0^2+\frac{K^2}{\varepsilon^3}\right)\|\nabla\mathbf{u}^{(n-1)}\|^2+\frac{C_1}{2}\|\Delta\mathbf{u}^{(n-1)}\|^2+C(\Omega,D,\omega_0,\mathbf{f}).$$

This is (almost) the same estimation as the last inequality in [40, p. 427]. Therefore, the rest of the proof follows the proof of [40, Theorem 3.1(ii)]. This results in the desired estimation (22).

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] 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.

[3] J. Bai and X.-C. Feng, *Fractional-order anisotropic diffusion for image denoising*, IEEE Trans. Image Process., 16 (2007), pp. 2492–2502.

[4] M. Bertalmío, G. Sapiro, V. Caselles, and C. Ballester, *Image inpainting*, in Proceedings of the ACM Computer Graphics and Interactive Techniques, SIGGRAPH '00, New York, NY, 2000, ACM Press/Addison–Wesley, pp. 417–424.

[5] A. Bertozzi, S. Esedoḡlu, and A. Gillette, *Analysis of a two-scale Cahn–Hilliard model for binary image inpainting*, Multiscale Model. Simul., 6 (2007), pp. 913–936.

[6] A. L. Bertozzi, S. Esedoḡlu, and A. Gillette, *Inpainting of binary images using the Cahn–Hilliard equation*, IEEE Trans. Image Process., 16 (2007), pp. 285–291.

[7] 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.

[8] 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.

[9] A. Bueno-Orovio, D. Kay, and K. Burrage, *Fourier spectral methods for fractional-in-space reaction-diffusion equations*, BIT, 54 (2014), pp. 937–954.

[10] 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.

[11] 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.

[12] T. Chan and J. Shen, *Variational restoration of nonflat image features: Models and algorithms*, SIAM J. Appl. Math., 61 (2001), pp. 1338–1361.

[13] T. F. Chan and J. Shen, *Nontexture inpainting by curvature-driven diffusions*, J. Vis. Commun. Image Represent., 12 (2001), pp. 436–449.

[14] ———, *Mathematical models for local nontexture inpaintings*, SIAM J. Appl. Math., 62 (2002), pp. 1019–1043.

[15] ———, *Image Processing and Analysis: Variational, PDE, Wavelet, and Stochastic Methods*, Other Titles in Applied Mathematics, SIAM, 2005.

[16] Y. Chen, Y. 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). Article Id 786281.

[17] L. Cherfils, H. Fakih, and A. Miranville, *On the Bertozzi–Esedoglu–Gillette–Cahn–Hilliard equation with logarithmic nonlinear terms*. 2014.

[18] 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.

[19] C. M. Elliott and A. M. Stuart, *The global dynamics of discrete semilinear parabolic equations*, SIAM J. Numer. Anal., 30 (1993), pp. 1622–1663.

[20] S. Esedoḡlu and J. Shen, *Digital inpainting based on the Mumford–Shah–Euler image model*, European J. Appl. Math., 13 (2002), pp. 353–370.

[21] D. J. Eyre, *Systems of Cahn–Hilliard equations*, SIAM J. Appl. Math., 53 (1993), pp. 1686–1712.

[22] ———, *An unconditionally stable one-step scheme for gradient systems*, Technical report, Department of Mathematics, University of Utah, Salt Lake City, UT, 1998.

[23] D. Garcia, *Robust smoothing of gridded data in one and higher dimensions with missing values*, Computat. Statist. Data Anal., 54 (2010), pp. 1167–1178.

[24] A. Goshtasby, D. A. Turner, and L. V. Ackerman, *Matching of tomographic slices for interpolation*, IEEE Trans. Med. Imag., 11 (1992), pp. 507–516.

[25] 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.

[26] 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.

[27] M. E. Hochstenbach and L. Reichel, *Fractional Tikhonov regularization for linear discrete ill-posed problems*, BIT, 51 (2011), pp. 197–215.

[28] 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.

[29] W. A. Kalender, R. Hebel, and J. Ebersberger, *Reduction of CT artifacts caused by metallic implants*, Radiology, 164 (1987), pp. 576–577.

[30] E. Klann and R. Ramlau, *Regularization by fractional filter methods and data smoothing*, Inverse Problems, 24 (2008), p. 025018.

[31] 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.

[32] 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.

[33] R. M. Lewitt and R. H. T. Bates, *Image reconstruction from projections: III: Projection completion methods (theory)*, Optik, 50 (1978), pp. 189–204.

[34] R. Metzler and J. Klafter, *The random walk's guide to anomalous diffusion: A fractional dynamics approach*, Phys. Rep., 339 (2000), pp. 1–77.

[35] 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., (2014), pp. 1–59.

[36] K. B. Oldham and J. Spanier, *The fractional calculus; Theory and applications of differentiation and integration to arbitrary order*, Mathematics in science and engineering : A series in monographs and textbooks 111, Elsevier, Burlington, MA, 1974.

[37] 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, AM, 2006, IEEE, pp. 129–138.

[38] S. Salsa, *Partial differential equations in action: From modelling to theory*, Universitext, Springer, Mailand, 2009.

[39] S. G. Samko, A. A. Kilbas, and O. I. Marichev, *Fractional Integrals and Derivatives : Theory and Applications*, Gordon and Breach, Yverdon, 1993.

[40] C.-B. Schönlieb and A. Bertozzi, *Unconditionally stable schemes for higher order inpainting*, Commun. Math. Sci., 9 (2011), pp. 413–457.

[41] J. Shen, S. H. Kang, and T. F. Chan, *Euler's elastica and curvature-based inpainting*, SIAM J. Appl. Math., 63 (2003), pp. 564–592.

[42] J. L. Troutman, *Variational calculus and optimal control. Optimization with elementary convexity*, Undergrad. Texts Math., Springer, New York, NY, 2nd ed., 1996.

[43] 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.

[44] 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.

[45] 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, 2001, IEEE, pp. 195–202.

[46] 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.

[47] 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.

[48] P. Yi-Fei, *Application of fractional differential approach to digital image processing*, J. Sichuan Univ. Eng. Sci. Ed., 3 (2007), p. 022.

[49] 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). Article ID 173748.

[50] 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.

[51] 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.