

## Fast Solvers for Cahn–Hilliard Inpainting\*

Jessica Bosch<sup>†</sup>, David Kay<sup>‡</sup>, Martin Stoll<sup>†</sup>, and Andrew J. Wathen<sup>§</sup>

**Abstract.** The solution of Cahn–Hilliard variational inequalities is of interest in many applications. We discuss the use of them as a tool for binary image inpainting. This has been done before using double-well potentials but not for nonsmooth potentials as considered here. The existing bound constraints are incorporated via the Moreau–Yosida regularization technique. We develop effective preconditioners for the efficient solution of the Newton steps associated with the fast solution of the Moreau–Yosida regularized problem. Numerical results illustrate the efficiency of our approach. Moreover, precise eigenvalue intervals are given for the preconditioned system using a double-well potential. A comparison between the smooth and nonsmooth Cahn–Hilliard inpainting models shows that the latter achieves better results.

**Key words.** Cahn–Hilliard equation, image inpainting, binary images, preconditioning, Schur complement approximation

**AMS subject classifications.** 68U90, 94A08, 65F08, 65F10, 65N22, 65F50, 65M60, 93C20, 35R35, 74S05, 35K55, 80A22

**DOI.** 10.1137/130921842

**1. Introduction.** Image inpainting is the process of filling in missing or damaged parts of images based on information from surrounding areas. Given an image  $f$ , defined on a domain  $\Omega \subset \mathbb{R}^2$ , the target is to reconstruct the original image  $u$  in the damaged domain  $D \subset \Omega$ .

Bertalmío et al. [4] introduced image inpainting for digital image processing. Their model is based on a third-order nonlinear PDE that attempts to replicate the basic techniques used by museum artists who specialize in restoration. After the user marks the regions to be inpainted, these areas are automatically and simultaneously filled in. The basic idea is the smooth propagation of the information from surrounding areas in the isophote (lines of equal gray values) direction (transport process). More precisely,  $u$  being a gray valued image to be inpainted in the region  $D$ , the PDE to solve is then

$$\partial_t u = \nabla^\perp u \cdot \nabla \Delta u \quad \text{in } D,$$

where  $\Delta u$  is a measure of image smoothness and  $\nabla^\perp u$  is the propagation direction, i.e., the direction of smallest spatial change. The operator  $\nabla^\perp$  denotes the perpendicular gradient

\*Received by the editors May 22, 2013; accepted for publication (in revised form) October 3, 2013; published electronically January 2, 2014. This research was based on work supported in part by Award No. KUK-C1-013-04, made by King Abdullah University of Science and Technology (KAUST).

<http://www.siam.org/journals/siims/7-1/92184.html>

<sup>†</sup>Computational Methods in Systems and Control Theory, Max Planck Institute for Dynamics of Complex Technical Systems, 39106 Magdeburg, Germany ([bosch@mpi-magdeburg.mpg.de](mailto:bosch@mpi-magdeburg.mpg.de), [stollm@mpi-magdeburg.mpg.de](mailto:stollm@mpi-magdeburg.mpg.de)). Parts of this work were performed while the first author was visiting the Oxford Centre for Collaborative Applied Mathematics (OCCAM), University of Oxford.

<sup>‡</sup>Oxford Department of Computer Science, University of Oxford, Oxford OX1 3QD, UK ([dkay@cs.ox.ac.uk](mailto:dkay@cs.ox.ac.uk)).

<sup>§</sup>Numerical Analysis Group, Mathematical Institute, Oxford OX1 3LB, UK ([Andy.Wathen@maths.ox.ac.uk](mailto:Andy.Wathen@maths.ox.ac.uk)).

$(-\partial_y, \partial_x)$ . This equation is combined with intermediate steps of anisotropic diffusion

$$\partial_t u = \phi_\epsilon(x) |\nabla u| \nabla \cdot \frac{\nabla u}{|\nabla u|},$$

in order to avoid the crossing of isophote lines inside  $D$ . Here,  $\phi_\epsilon$  is a smooth function that forces the equation to act only in  $D$ , and  $\nabla \cdot \frac{\nabla u}{|\nabla u|}$  is the curvature along isophotes.

Another perspective on this model shows the connections with the Navier–Stokes equation of fluid dynamics; see [3]. It turns out that the steady state equation of the above model in [4],

$$\nabla^\perp u \cdot \nabla \Delta u = 0,$$

is exactly the equation satisfied by steady state inviscid flows in the two-dimensional incompressible Navier–Stokes model, in which the image intensity function  $u$  plays the role of the stream function. Moreover, the smoothness  $w := \Delta u$  corresponds to the vorticity and the isophote direction  $v := \nabla^\perp u$  becomes the fluid velocity. Then, the smoothness estimator is now evolved according to

$$\partial_t w + v \cdot \nabla w = \nu \nabla \cdot [g(|\nabla w|) \nabla w],$$

where  $\nu$  is a constant for the fluid viscosity (anisotropic diffusion in the inpainting sense), and  $g$  allows for anisotropic diffusion of  $w$  within the inpainting region only.

A number of variational- and PDE-based approaches have been considered. Chan and Shen [17, 14] study a second-order variational method, the so-called total variation (TV) inpainting model, that is closely connected to the classical TV denoising model of Rudin and Osher [50] and Rudin, Osher, and Fatemi [51]. The TV inpainting model is to minimize

$$(1.1) \quad E_{\text{TV}}(u) = \int_{\Omega} |\nabla u| \, dx + \frac{\omega}{2} \int_{\Omega \setminus D} (u - f)^2 \, dx,$$

where  $\omega$  is a fitting constant or Lagrange multiplier. The left integral in (1.1) is the regularization term which penalizes here the length of isophote lines. The right integral is the so-called fidelity term of the inpainting approach, which keeps the solution  $u$  close to the given image  $f$  in  $\Omega \setminus D$ . The Euler–Lagrange equation of (1.1) is then

$$\partial_t u = \nabla \cdot \frac{\nabla u}{|\nabla u|} + \omega(f - u) \chi_{\Omega \setminus D}.$$

This model successfully propagates sharp edges into the damaged domain. However, it can neither connect contours across very large distances nor preserve curvatures.

To overcome the former drawback of the TV inpainting model, Chan and Shen [16] proposed a variational third-order approach based on curvature driven diffusion (CDD). It is governed by

$$\partial_t u = \nabla \cdot \left[ \frac{G(\kappa, x)}{|\nabla u|} \nabla u \right] + \omega(f - u) \chi_{\Omega \setminus D},$$

where  $\kappa$  is the curvature  $\nabla \cdot \frac{\nabla u}{|\nabla u|}$  and  $G(\kappa, x)$  is the diffusion coefficient in order to penalize large curvature.

Inspired by the elastica curve model [42] (see also the proposal of Masnou and Morel [41]), Chan, Kang, and Shen [15] and Chan and Shen [18] proposed minimizing

$$E_{EE}(u) = \int_{\Omega} (\alpha + \beta\kappa^2) |\nabla u| \, dx + \frac{\omega}{2} \int_{\Omega \setminus D} (u - f)^2 \, dx,$$

with positive weights  $\alpha$  and  $\beta$ . This so-called Euler elastica model is of fourth order and overcomes the drawbacks of the third-order inpainting models [4, 16]. The regularization term here penalizes not only the length of isophote lines (as is the case in the TV inpainting method) but also their curvature. Euler's elastica model combines CDD and the transport process of Bertalmío et al. [4] and results in a smooth continuation of isophote lines over the inpainting domain also over large distances.

The Mumford–Shah inpainting model first appeared in Tsai, Yezzi, and Willsky [57] as an active contour method and in Chan and Shen [17] as an alternative to the TV inpainting approach. The model is to minimize

$$E_{MS}(u, \Gamma) = \frac{\gamma}{2} \int_{\Omega \setminus \Gamma} |\nabla u|^2 \, dx + \frac{\omega}{2} \int_{\Omega \setminus D} (u - f)^2 \, dx + \alpha \text{length}(\Gamma),$$

where  $\gamma$ ,  $\omega$ , and  $\alpha$  are positive weights and  $\Gamma$  denotes the edge collection. If  $D$  is empty, then the model is exactly the classical Mumford–Shah denoising and segmentation model [43]. Like the TV inpainting approach, the Mumford–Shah inpainting model is of second order and cannot connect isophotes across large distances.

In order to improve this method, Esedoğlu and Shen [26] proposed the Mumford–Shah–Euler inpainting model, where the length energy is upgraded to Euler's elastica energy. The energy to be minimized is then

$$E_{MSE}(u, \Gamma) = \frac{\gamma}{2} \int_{\Omega \setminus \Gamma} |\nabla u|^2 \, dx + \frac{\omega}{2} \int_{\Omega \setminus D} (u - f)^2 \, dx + \int_{\Gamma} (\alpha + \beta\kappa^2) \, ds,$$

leading to a fourth-order nonlinear PDE.

A new approach in the class of fourth-order inpainting is the application of a modified Cahn–Hilliard equation, proposed by Bertozzi, Esedoğlu, and Gillette [6, 5]. The Cahn–Hilliard equation is a PDE of fourth order, which is used in materials science [45, 32], image processing [21], and chemistry [62]. It was originally introduced to model phase separation in binary alloys [37, 13] and follows the evolution of a function  $u(x)$ , known as the order parameter, which smoothly varies between the values 0 and 1 across an interface. This represents the two different phases, i.e., black and white (as in this paper). Phase separation is modeled either by a smooth free energy, e.g., double-well potentials [22] such as

$$(1.2) \quad \psi_s(u) := u^2(u - 1)^2$$

and logarithmic potentials [19], or by nonsmooth double obstacle potentials [7, 8] such as

$$(1.3) \quad \psi_{ns}(u) := \begin{cases} \frac{1}{2}u(1 - u), & 0 \leq u \leq 1, \\ \infty & \text{otherwise.} \end{cases}$$

The Cahn–Hilliard equation combined with an additional fidelity term acting on the known parts of an image provides a nice tool for binary image inpainting. The smoothing property of the standard Cahn–Hilliard equation allows a natural connection of the contours across damaged parts. In particular, one can control this connection subject to the size of the damaged region by the interfacial parameter  $\varepsilon$ . The additional fidelity term keeps the new image close to the original one in those parts where picture information is available.

So far, we have observed only the use of double-well potentials for Cahn–Hilliard inpainting [33, 5, 12]. We additionally consider the nonsmooth one in (1.3) and will show that qualitatively better results can be obtained using the latter; see section 7.4. Moreover, using this potential, we can control the restriction  $0 \leq u \leq 1$  more precisely. All in all, we consider the efficient solution of Cahn–Hilliard inpainting with respect to preconditioning for both types of potentials. Another new feature of our approach is full automation in the sense that there is no longer a need for the user to mark the regions to be inpainted. Given tolerances  $tol_{\text{low}}$  and  $tol_{\text{up}}$ , the algorithm marks all points  $x \in \Omega$ , which fulfill the conditions  $u(x) > tol_{\text{low}}$  and  $u(x) < tol_{\text{up}}$ .

For image inpainting, efficient numerical schemes for higher-order methods are an active area of research. As discussed in [18], one of the most interesting open problems in digital inpainting is, in fact, the fast and exact digital realization. In the case of Cahn–Hilliard inpainting, Bertozzi, Esedoğlu, and Gillette [6, 5] proposed a semi-implicit time-discrete scheme, the so-called convexity splitting [60, 55], which is guaranteed to be unconditionally stable [6]. The convexity splitting was originally introduced by Elliott and Stuart [23] and is often attributed to Eyre [27, 28]. We apply this idea whereupon we work with a mixed formulation of the fourth-order system; i.e., we deal with two coupled second-order equations. For the space discretization, a common way is the use of a fast Fourier transform (FFT) method to compute the finite differences for the derivatives; see, e.g., [33, 6, 12]. In contrast, we propose the use of finite elements, as adopted in our previous work [10]. This typically allows us to compute missing information on arbitrary domains, like complex three-dimensional (3D) objects. Another motivation is the fact that an efficient FFT-based implementation employing the nonsmooth potential would typically suffer from the arising nonconstant coefficient; see section 7.2. Using the nonsmooth potential, we additionally have to take care of the nonsmoothness and nonlinearity, for which we suggest a semismooth Newton method [38, 58, 40, 10].

As we show in the course of this paper, the solution of a linear system  $Ax = b$  with  $A$  being a real, sparse matrix is at the heart of our method for both variants, using the smooth and nonsmooth potentials. For huge linear systems, the application of direct solvers such as UMFPACK [20] becomes infeasible. As a result, iterative methods have to be employed (see, e.g., [34, 53] for introductions to this field). We propose the use of Krylov subspace solvers. The convergence behavior of the iterative scheme typically depends on the conditioning of the problem and the clustering of the eigenvalues. These properties can be enhanced using preconditioning techniques  $P^{-1}Ax = P^{-1}b$ , where  $P$  is an invertible matrix that is easy to invert and resembles  $A$ . In this paper, we provide an efficient preconditioner  $P$  for the solution of the smooth and nonsmooth modified Cahn–Hilliard equations using an effective Schur complement approximation and (algebraic) multigrid developed for elliptic systems [29, 53, 52]. Moreover, precise eigenvalue bounds are given for the preconditioned system using the double-well potential.

The paper is organized as follows. In section 2, we derive the time-discrete numerical Cahn–Hilliard inpainting scheme for the nonsmooth potential (the smooth version with the double-well potential follows analogously). We consider in section 3 a semismooth Newton (SSN) method to solve the regularized subproblems and show superlinear convergence in function space. We derive the linear systems arising from a discretization using finite elements in section 4. In sections 5–6, we analyze the linear systems for the double-well as well as double obstacle potentials and propose preconditioning strategies. Section 7 illustrates the competitiveness of our preconditioners for both problem setups. Section 8 summarizes our findings.

**2. The modified Cahn–Hilliard equation.** The Cahn–Hilliard equation is derived as the  $H^{-1}$ -gradient flow of the Ginzburg–Landau energy

$$E(u) = \int_{\Omega} \frac{\gamma\varepsilon}{2} |\nabla u|^2 + \frac{1}{\varepsilon} \psi(u) \, dx,$$

where  $\varepsilon > 0$  is proportional to the thickness of the interfacial region and  $\gamma > 0$  is a constant related to the interfacial energy density. As mentioned in the introduction,  $\psi(u)$  can be the double-well potential in (1.2) (see, e.g., [22]) or the nonsmooth double obstacle potential in (1.3) as in [7, 8]. Existence and uniqueness of weak solutions were shown in [24, 7] for both problem setups.

Now, we formulate the modified version of the nonsmooth Cahn–Hilliard equation for the inpainting problem. Let  $f(x)$  be a given binary image in a domain  $\Omega$ , and let  $D \subset \Omega$  be the inpainting domain (damaged or missing parts). Then the variable  $u(x, t)$  evolves in time to become a fully inpainted version of  $f(x)$  under the following system:

$$(2.1) \quad \partial_t u = -\Delta \left( \gamma\varepsilon \Delta u - \frac{1}{\varepsilon} (\psi'_0(u) + \mu) \right) + \omega(x)(f - u),$$

$$(2.2) \quad \mu \in \partial\beta_{[0,1]}(u),$$

$$(2.3) \quad 0 \leq u \leq 1,$$

$$(2.4) \quad \frac{\partial u}{\partial n} = \frac{\partial \Delta u}{\partial n} = 0 \quad \text{on } \partial\Omega,$$

where

$$(2.5) \quad \omega(x) = \begin{cases} 0 & \text{if } x \in D, \\ \omega_0 & \text{if } x \in \Omega \setminus D. \end{cases}$$

Here we have written  $\psi_{ns}$  in (1.3) via the introduced indicator function

$$I_{[0,1]}(u) := \begin{cases} 0, & 0 \leq u \leq 1, \\ \infty & \text{otherwise} \end{cases}$$

as

$$\psi_{ns}(u) = \psi_0(u) + I_{[0,1]}(u),$$

where  $\psi_0(u) = \frac{1}{2}u(1-u)$  and  $\partial\beta_{[0,1]}(u)$  denotes the subdifferential of the nonsmooth part  $\beta_{[0,1]}(u) := \int_{\Omega} I_{[0,1]}(u)$  of the energy  $E$ . The system (2.1)–(2.4) is identical to the standard

Cahn–Hilliard equation except for the second term on the right-hand side of (2.1), which is the so-called fidelity term that keeps the solution close to the given image  $f(x)$  in those areas where image information is available.

*Remark 2.1.* The main difficulty of the modified Cahn–Hilliard system lies in the convergence analysis. The corresponding evolution equations do not follow a variational principle, in contrast to the original Cahn–Hilliard model. In particular, this forms the main difference between this and our previous work [10]. In [10], the appropriate minimization problem under an implicit time discretization is additionally analyzed. For the problem considered in this work, special methods concerning the absence of a variational principle (see, e.g., [55]) are applied.

The new modified Cahn–Hilliard equation is not strictly a gradient flow. As mentioned in the beginning, the original Cahn–Hilliard equation, (2.1)–(2.4) with  $\omega \equiv 0$ , is indeed the  $H^{-1}$ -gradient flow of the energy  $E(u)$ , while the fidelity term in (2.1) is derived as the  $L^2$ -gradient flow of the energy

$$(2.6) \quad E_2(u) = \frac{1}{2} \int_{\Omega} \omega (f - u)^2 \, dx.$$

But in total, the modified Cahn–Hilliard equation is neither a gradient flow in  $H^{-1}$  nor in  $L^2$ . In [6], the authors propose the application of convexity splitting to these energy functionals leading to an efficient numerical scheme which is unconditionally gradient stable.

Before we apply convexity splitting, we handle the pointwise constraints in (2.3) with a Moreau–Yosida regularization technique, as motivated in [10, 38]. Instead of the energy functional

$$(2.7) \quad E(u) = \int_{\Omega} \frac{\gamma\varepsilon}{2} |\nabla u|^2 + \frac{1}{\varepsilon} (\psi_0(u) + I_{[0,1]}(u)) \, dx$$

we consider

$$E_1(u_\nu) = \int_{\Omega} \frac{\gamma\varepsilon}{2} |\nabla u_\nu|^2 + \frac{1}{\varepsilon} \psi_0(u_\nu) + \frac{1}{2\nu} |\max(0, u_\nu - 1)|^2 + \frac{1}{2\nu} |\min(0, u_\nu)|^2 \, dx,$$

such that we obtain

$$(2.8) \quad \partial_t u_\nu = -\Delta \left( \gamma\varepsilon \Delta u_\nu - \frac{1}{\varepsilon} \psi'_0(u_\nu) - \theta_\nu(u_\nu) \right) + \omega(x)(f - u_\nu),$$

$$(2.9) \quad \frac{\partial u_\nu}{\partial n} = \frac{\partial \Delta u_\nu}{\partial n} = 0 \quad \text{on } \partial\Omega,$$

where

$$\theta_\nu(u_\nu) := \frac{1}{\nu} \max(0, u_\nu - 1) + \frac{1}{\nu} \min(0, u_\nu)$$

and  $0 < \nu \ll 1$  denotes the regularization/penalty parameter. Without doing so, the system (2.1)–(2.4) leads to a variational inequality, which is known to be handled by introducing Lagrange multipliers associated with the constraints in (2.3). But their low regularity—since  $u \in H^1(\Omega)$ , these multipliers are measures (elements of  $H^1(\Omega)^*$ )—complicates the numerical

treatment, and therefore we follow the idea of Moreau–Yosida regularization proposed in [38]. Another motivation for this procedure emerges in the preconditioning of the arising linear system of equations, which leads us to better convergence rates with the regularized treatment; see more details in [9].

In the convexity splitting scheme [27, 28, 23, 60, 33, 5, 12], the energy functionals are divided into two parts, respectively—a convex plus a concave one. The convex part is then treated implicitly whilst the concave part is treated explicitly. In our case, we split  $E_1(u_\nu)$  as  $E_1(u_\nu) = E_{11}(u_\nu) - E_{12}(u_\nu)$ , where

$$E_{11}(u_\nu) = \int_{\Omega} \frac{\gamma\varepsilon}{2} |\nabla u_\nu|^2 + \frac{C_1}{2} |u_\nu|^2 + \frac{1}{2\nu} |\max(0, u_\nu - 1)|^2 + \frac{1}{2\nu} |\min(0, u_\nu)|^2 dx,$$

$$E_{12}(u_\nu) = \int_{\Omega} -\frac{1}{\varepsilon} \psi_0(u_\nu) + \frac{C_1}{2} |u_\nu|^2 dx,$$

as well as  $E_2(u_\nu) = E_{21}(u_\nu) - E_{22}(u_\nu)$ , where

$$E_{21}(u_\nu) = \int_{\Omega} \frac{C_2}{2} |u_\nu|^2 dx,$$

$$E_{22}(u_\nu) = \int_{\Omega} -\frac{\omega}{2} (f - u_\nu)^2 + \frac{C_2}{2} |u_\nu|^2 dx.$$

The constants  $C_1$  and  $C_2$  are positive and need to be chosen large enough such that the energies  $E_{11}(u_\nu)$ ,  $E_{12}(u_\nu)$ ,  $E_{21}(u_\nu)$ , and  $E_{22}(u_\nu)$  are convex. Therefore, we have to choose  $C_1 > 0$  and  $C_2 > \omega_0$ .

*Remark 2.2.* Note that we do not need the  $C_1$ -term, as  $E_{12}(u_\nu)$  is already convex without it. But with respect to the preconditioning we obtain better results by keeping this term; see section 7.3.

For this splitting, together with the backward Euler discretization for the time derivative  $\partial_t u_\nu$ , the resulting time-stepping scheme is

$$\frac{u_\nu^{(n)} - u_\nu^{(n-1)}}{\tau} = -\Delta_{H^{-1}}(E_{11}(u_\nu^{(n)}) - E_{12}(u_\nu^{(n-1)})) - \Delta_{L^2}(E_{21}(u_\nu^{(n)}) - E_{22}(u_\nu^{(n-1)})),$$

where  $\Delta_{H^{-1}}$  and  $\Delta_{L^2}$  represent the gradient descent with respect to the  $H^{-1}$ - and  $L^2$ -inner products, 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

$$(2.10) \quad \frac{u_\nu^{(n)} - u^{(n-1)}}{\tau} + \gamma\varepsilon\Delta^2 u_\nu^{(n)} - C_1\Delta u_\nu^{(n)} + C_2u_\nu^{(n)} - \Delta\theta_\nu(u_\nu^{(n)})$$

$$= \frac{1}{\varepsilon}\Delta(\psi'_0(u^{(n-1)})) + \omega(f - u^{(n-1)}) - C_1\Delta u^{(n-1)} + C_2u^{(n-1)}.$$

By operator splitting, we obtain the weak formulation of (2.10) as

$$(2.11) \quad \left(\frac{1}{\tau} + C_2\right) (u_\nu, v) + (\nabla w_\nu, \nabla v) + C_1(\nabla u_\nu, \nabla v)$$

$$= (\omega(f - u^{(n-1)}), v) + C_1(\nabla u^{(n-1)}, \nabla v) + \left(\frac{1}{\tau} + C_2\right) (u^{(n-1)}, v),$$

$$(2.12) \quad (w_\nu, v) - \gamma\varepsilon(\nabla u_\nu, \nabla v) - (\theta_\nu(u_\nu), v) = \frac{1}{\varepsilon}(\psi'_0(u^{(n-1)}), v)$$

for all  $v \in H^1(\Omega)$ , where  $u_\nu = u_\nu^{(n)}$  and  $w_\nu = w_\nu^{(n)}$ . In the following,  $(\cdot, \cdot)$  and  $\langle \cdot, \cdot \rangle$  stand for the  $L^2(\Omega)$ -inner product and the duality pairing of  $H^1(\Omega)$  and  $H^1(\Omega)^*$ , respectively.

**Remark 2.3.** Using the double-well potential, the authors proved in [5] global existence and uniqueness of a weak solution of the modified continuous fourth-order system. Moreover, in [12], the existence of a weak solution of the stationary equation for this system is shown. The application of the convexity splitting scheme leads to the conditions  $C_1 > \frac{1}{\varepsilon}$  and  $C_2 > \omega_0$ ; see [6]. In [12], the authors proved that this time-stepping scheme is unconditionally stable in the sense that the numerical solution  $u^{(n)}$  is uniformly bounded on a finite time interval. Moreover, the discrete solution converges to the exact solution of the continuous fourth-order system as  $\tau \rightarrow 0$ . These properties make the time-discrete scheme a stable and reliable discrete approximation of the continuous fourth-order equation.

**Remark 2.4.** In the following, the weak formulation in (2.11)–(2.12) will be treated as in our previous work [10]. As mentioned in the beginning, the main difference lies in the additional fidelity term that we have in this work. It especially makes the construction of efficient preconditioners more complicated. Moreover, the preconditioned Cahn–Hilliard inpainting model using a double-well potential is analyzed in this work.

**3. SSN method.** We apply the function space-based algorithm motivated in [38, 10] for solving the nonsmooth time-discrete Cahn–Hilliard problem. For a specified sequence  $\nu \rightarrow 0$  we solve the optimality system (2.11)–(2.12), compactly written as

$$(3.1) \quad F_\nu(u_\nu, w_\nu) = (F_\nu^{(1)}(u_\nu, w_\nu), F_\nu^{(2)}(u_\nu, w_\nu)) = 0,$$

for every  $\nu$  by an SSN algorithm. In (3.1), the components are defined by

$$\begin{aligned} \langle F_\nu^{(1)}(u, w), v \rangle &= \left( \frac{1}{\tau} + C_2 \right) (u, v) + (\nabla w, \nabla v) + C_1(\nabla u, \nabla v) \\ &\quad - (\omega(f - u^{(n-1)}), v) - C_1(\nabla u^{(n-1)}, \nabla v) - \left( \frac{1}{\tau} + C_2 \right) (u^{(n-1)}, v), \\ \langle F_\nu^{(2)}(u, w), v \rangle &= (w, v) - \gamma\varepsilon(\nabla u, \nabla v) - (\theta_\nu(u), v) - \frac{1}{\varepsilon}(\psi'_0(u^{(n-1)}), v) \end{aligned}$$

for all  $u, w, v \in H^1(\Omega)$ . Due to the presence of the max- and min-operators in the definition of  $\theta_\nu$ ,  $F_\nu$  is not Fréchet-differentiable. However, it satisfies the weaker notion of Newton differentiability; see [39, 38].

**Definition 3.1.** Let  $X$  and  $Z$  be Banach spaces, and let  $E \subset X$  be an open subset. A mapping  $F: E \rightarrow Z$  is called Newton-differentiable in  $U \subset E$  if there exists a family of mappings  $G: U \rightarrow Z$  such that

$$\lim_{h \rightarrow 0} \frac{\|F(x+h) - F(x) - G(x+h)h\|_Z}{\|h\|_X} = 0 \quad \forall x \in U.$$

The operator  $G$  is called a Newton derivative of  $F$  on  $U$ .

For such mappings, the following convergence result for the (semismooth) Newton iteration

$$(3.2) \quad x^{(k+1)} = x^{(k)} - G(x^{(k)})^{-1}F(x^{(k)}), \quad k = 0, 1, \dots,$$

holds true, where  $G$  is a Newton derivative of  $F$ . For its proof we refer the reader to Theorem 1.1 in [39].

**Theorem 3.2.** *Let  $x^*$  be a solution of  $F(x) = 0$ , and suppose that  $F: E \subset X \rightarrow Z$  is Newton-differentiable in a neighborhood  $U$  of  $x^*$  with  $\{\|G(x)^{-1}\|_{\mathcal{L}(Z,X)} : x \in U\}$  bounded. Then the sequence  $\{x^{(k)}\}_{k \in \mathbb{N}}$  generated by (3.2) converges superlinearly to  $x^*$  provided that  $\|x^{(0)} - x^*\|_X$  is sufficiently small.*

**Lemma 3.3.** *The mapping  $F_\nu: H^1(\Omega) \times H^1(\Omega) \rightarrow H^1(\Omega)^* \times H^1(\Omega)^*$  is Newton-differentiable. Furthermore, the operator  $G_\nu(u, w)$  given by*

$$\langle G_\nu(u, w)(\delta u, \delta w), (\phi, \psi) \rangle = \begin{pmatrix} (\frac{1}{\tau} + C_2)(\delta u, \phi) + (\nabla \delta w, \nabla \phi) + C_1(\nabla \delta u, \nabla \phi) \\ (\delta w, \psi) - \gamma \varepsilon (\nabla \delta u, \nabla \psi) - \frac{1}{\nu} (\chi_{\mathcal{A}(u)} \delta u, \psi) \end{pmatrix}$$

serves as a Newton derivative for  $F_\nu$ , where  $\chi_{\mathcal{A}(u)}$  is the characteristic function of the set

$$\mathcal{A}(u) := \{x \in \Omega : u(x) > 1 \text{ or } u(x) < 0\}.$$

For the proof we refer the reader to Lemma 5.3 in [38] and Proposition 4.1 in [39]. Because of the explicit treatment of the additional term  $(\omega(f - u^{(n-1)}))$  we do not have to take care of it.

**Lemma 3.4.** *The SSN method (3.2) (with  $F$  and  $G$  replaced by  $F_\nu$  and  $G_\nu$ ) converges superlinearly to  $(u_\nu, w_\nu)$ , the solution of (3.1), provided that  $\|(u^{(0)}, w^{(0)}) - (u_\nu, w_\nu)\|_{H^1(\Omega) \times H^1(\Omega)}$  is sufficiently small.*

*Proof.* First of all, we show that  $G_\nu(u, w)$  is invertible for all  $(u, w) \in H^1(\Omega) \times H^1(\Omega)$ , i.e., that a unique solution  $(\delta u, \delta w) \in H^1(\Omega) \times H^1(\Omega)$  of the following linear system exists:

$$(3.3) \quad \left(\frac{1}{\tau} + C_2\right) (\delta u, \phi) + (\nabla \delta w, \nabla \phi) + C_1(\nabla \delta u, \nabla \phi) = 0 \quad \forall \phi \in H^1(\Omega),$$

$$(3.4) \quad (\delta w, \phi) - \gamma \varepsilon (\nabla \delta u, \nabla \phi) - \frac{1}{\nu} (\chi_{\mathcal{A}(u)} \delta u, \phi) = 0 \quad \forall \phi \in H^1(\Omega).$$

Multiplying (3.4) with  $-(\frac{1}{\tau} + C_2)$ , testing (3.3) with  $\delta w$  and (3.4) with  $\delta u$ , and adding the two equations, we obtain

$$(3.5) \quad 0 = \|\nabla \delta w\|^2 + C_1(\nabla \delta u, \nabla \delta w) + \gamma \varepsilon \left(\frac{1}{\tau} + C_2\right) \|\nabla \delta u\|^2 + \frac{1}{\nu} \left(\frac{1}{\tau} + C_2\right) (\chi_{\mathcal{A}(u)} \delta u, \delta u).$$

Applying Young’s inequality with  $\alpha > 0$  to

$$(3.6) \quad (\nabla \delta u, \nabla \delta w) \geq -\alpha \|\nabla \delta u\|^2 - \frac{1}{4\alpha} \|\nabla \delta w\|^2$$

yields in (3.5)

$$0 \geq \left(1 - \frac{C_1}{4\alpha}\right) \|\nabla \delta w\|^2 + \left(\gamma \varepsilon \left(\frac{1}{\tau} + C_2\right) - C_1\alpha\right) \|\nabla \delta u\|^2 + \frac{1}{\nu} \left(\frac{1}{\tau} + C_2\right) (\chi_{\mathcal{A}(u)} \delta u, \delta u) \geq 0,$$

where we choose  $\alpha \in (0, \frac{\gamma\varepsilon(\frac{1}{\tau} + C_2)}{C_1})$ . We obtain  $\|\nabla\delta w\|^2 = \|\nabla\delta u\|^2 = 0$ , which implies that  $\delta u$  and  $\delta w$  are constant. Then (3.3) gives  $\delta u = 0$  and finally (3.4) results in  $\delta w = 0$ . We have proved that  $G_\nu(u, w)$  is invertible for all  $(u, w) \in H^1(\Omega) \times H^1(\Omega)$ . Now, we show the boundedness of  $\|G_\nu^{-1}(u, w)\|_{\mathcal{L}((H^1(\Omega))^*, (H^1(\Omega))^2)}$  for all  $(u, w) \in H^1(\Omega) \times H^1(\Omega)$ .

$G_\nu(u, w)$  is invertible for all  $(u, w) \in H^1(\Omega) \times H^1(\Omega)$ ; i.e., for given  $(y_1, y_2) \in H^1(\Omega)^* \times H^1(\Omega)^*$ , there exists a unique pair  $(\delta u, \delta w) \in H^1(\Omega) \times H^1(\Omega)$  such that

$$(3.7) \quad \left(\frac{1}{\tau} + C_2\right) (\delta u, \phi) + (\nabla\delta w, \nabla\phi) + C_1(\nabla\delta u, \nabla\phi) = \langle y_1, \phi \rangle \quad \forall \phi \in H^1(\Omega),$$

$$(3.8) \quad (\delta w, \psi) - \gamma\varepsilon(\nabla\delta u, \nabla\psi) - \frac{1}{\nu}(\chi_{\mathcal{A}(u)}\delta u, \psi) = \langle y_2, \psi \rangle \quad \forall \psi \in H^1(\Omega)$$

is satisfied. Taking  $(\phi, \psi) = (\delta w, \delta u)$  in (3.7)–(3.8) and adding the two equations, we obtain

$$\begin{aligned} \gamma\varepsilon \left(\frac{1}{\tau} + C_2\right) \|\nabla\delta u\|^2 + \|\nabla\delta w\|^2 &= \langle y_1, \delta w \rangle + \langle y_2, \delta u \rangle - C_1(\nabla\delta u, \nabla\delta w) - \frac{1}{\nu} \underbrace{(\chi_{\mathcal{A}(u)}\delta u, \delta u)}_{\geq 0} \\ &\leq \langle y_1, \delta w \rangle + \langle y_2, \delta u \rangle - C_1(\nabla\delta u, \nabla\delta w). \end{aligned}$$

Using (3.6) with  $\alpha \in (0, \frac{\gamma\varepsilon(\frac{1}{\tau} + C_2)}{C_1})$ , we get

$$(3.9) \quad \underbrace{\left(\gamma\varepsilon \left(\frac{1}{\tau} + C_2\right) - C_1\alpha\right)}_{\geq 0} \|\nabla\delta u\|^2 + \underbrace{\left(1 - \frac{C_1}{4\alpha}\right)}_{\geq 0} \|\nabla\delta w\|^2 \\ \leq \langle y_1, \delta w \rangle + \langle y_2, \delta u \rangle \\ \leq \frac{1}{4\beta_1} \left(\|\delta w\|^2 + \|\nabla\delta w\|^2\right) + \frac{1}{4\beta_2} \left(\|\delta u\|^2 + \|\nabla\delta u\|^2\right) \\ + C \left(\|y_1\|_{H^1(\Omega)^*}^2 + \|y_2\|_{H^1(\Omega)^*}^2\right),$$

where we used the Cauchy–Schwarz and Young inequalities for the last result and the fact that the constant  $C > 0$  does not depend on  $\delta u$  or  $\delta w$ . Taking  $(\phi, \psi) = (1, 1)$  in (3.7)–(3.8), we get

$$(3.10) \quad (\delta u, 1) = \frac{1}{\frac{1}{\tau} + C_2} \langle y_1, 1 \rangle,$$

$$(3.11) \quad (\delta w, 1) = \frac{1}{\nu} (\chi_{\mathcal{A}(u)}\delta u, 1) - \frac{1}{\frac{1}{\tau} + C_2} \langle y_2, 1 \rangle.$$

From (3.10)–(3.11) and (3.9) we obtain by using the Poincaré–Friedrichs inequality and again Young’s inequality that

$$\|(\delta u, \delta w)\|_{H^1(\Omega) \times H^1(\Omega)} \leq C \left(\|y_1\|_{H^1(\Omega)^*} + \|y_2\|_{H^1(\Omega)^*}\right).$$

For  $\max(\|y_1\|_{H^1(\Omega)^*}, \|y_2\|_{H^1(\Omega)^*}) \leq \beta$  for some constant  $\beta > 0$ , we consequently have

$$\|G_\nu^{-1}(u, w)\|_{\mathcal{L}((H^1(\Omega)^*)^2, (H^1(\Omega))^2)} \leq \hat{C} \quad \forall (u, w) \in H^1(\Omega) \times H^1(\Omega)$$

with some constant  $\hat{C} > 0$  possibly depending on  $\varepsilon, \gamma, \tau, \nu, \beta_1, \beta_2, \beta_3$ , or  $\beta$  but not on  $u$  or  $w$ . Thus,  $F_\nu$  with associated Newton derivative  $G_\nu$  fulfills the conditions of Theorem 3.2, which completes the proof. ■

We have shown local superlinear convergence in function space of the SSN method for solving the regularized subproblem (3.1). We now want to discretize the problem in space and then discuss its efficient solution.

**4. Finite-element approximation.** We discretize (3.1) by finite elements [56]. In the following, we assume for simplicity that  $\Omega$  is a polyhedral domain. Generalizations to curved domains are possible using boundary finite elements with curved faces. Let  $\{\mathcal{R}_h\}_{h>0}$  be a triangulation of  $\Omega$  into disjoint open rectangular elements. The use of rectangles is motivated by performing the implementation with deal.II [1, 2]. Furthermore, we define  $\mathcal{R}_h$  to have maximal element size  $h := \max_{R \in \mathcal{R}_h} \{\text{diam}(R)\}$  and we set  $J_h$  to be the set of nodes of  $\mathcal{R}_h$  and  $p_j \in J_h$  to be the coordinates of these nodes. We approximate the infinite-dimensional space  $H^1(\Omega)$  by the finite-dimensional space

$$S_h := \{\phi \in C^0(\bar{\Omega}) : \phi|_R \in Q_1(R) \quad \forall R \in \mathcal{R}_h\} \subset H^1(\Omega)$$

of continuous, piecewise multilinear functions, where, e.g., for dimension 2 we have that  $Q_1 = \text{span}\{x^{\alpha_i}y^{\alpha_i} : \alpha_i \in \{0, 1\}, i = 1, 2\}$  are bilinear. With each  $p_j \in J_h$  we associate the nodal basis function  $\varphi_j \in S_h$  with the property  $\varphi_j(p_i) = \delta_{ij}$ ,  $i, j = 1, \dots, N$ .

The discretized version of the penalized problem (3.1) consists in finding  $(u_{\nu,h}, w_{\nu,h}) \in S_h \times S_h$  such that

$$(4.1) \quad \langle F_{\nu,h}^{(1)}(u_{\nu,h}, w_{\nu,h}), v_h \rangle = 0 \quad \forall v_h \in S_h,$$

$$(4.2) \quad \langle F_{\nu,h}^{(2)}(u_{\nu,h}, w_{\nu,h}), v_h \rangle = 0 \quad \forall v_h \in S_h,$$

where the components are

$$\begin{aligned} \langle F_{\nu,h}^{(1)}(u_{\nu,h}, w_{\nu,h}), v_h \rangle &= \left(\frac{1}{\tau} + C_2\right) (u_{\nu,h}, v_h)_h + (\nabla w_{\nu,h}, \nabla v_h) + C_1(\nabla u_{\nu,h}, \nabla v_h) \\ &\quad - (\omega(f_h - u_h^{(n-1)}), v_h)_h - C_1(\nabla u_h^{(n-1)}, \nabla v_h) - \left(\frac{1}{\tau} + C_2\right) (u_h^{(n-1)}, v_h)_h, \end{aligned}$$

$$\langle F_{\nu,h}^{(2)}(u_{\nu,h}, w_{\nu,h}), v_h \rangle = (w_{\nu,h}, v_h)_h - \gamma\varepsilon(\nabla u_{\nu,h}, \nabla v_h) - (\theta_\nu(u_{\nu,h}), v_h)_h - \frac{1}{\varepsilon}(\psi'_0(u_h^{(n-1)}), v_h)_h.$$

The semi-inner product  $(\cdot, \cdot)_h$  on  $C_0(\bar{\Omega})$  is defined by

$$(g_1, g_2)_h := \int_{\Omega} \pi_h(g_1(x)g_2(x)) \, dx = \sum_{i=1}^N (1, \varphi_i)g_1(p_i)g_2(p_i) \quad \forall g_1, g_2 \in C_0(\bar{\Omega}),$$

where  $\pi_h : C_0(\bar{\Omega}) \rightarrow S_h$  is the Lagrange interpolation operator. Within our finite-element framework, for a given  $(u_h, w_h) \in S_h \times S_h$ , every step of the SSN method for solving (4.1)–(4.2) requires computing  $(\delta u_h, \delta w_h) \in S_h \times S_h$  satisfying

$$(4.3) \quad \left( \frac{1}{\tau} + C_2 \right) (\delta u_h, v_h)_h + (\nabla \delta w_h, \nabla v_h) + C_1 (\nabla \delta u_h, \nabla v_h) = -F_{\nu, h}^{(1)}(u_h, w_h),$$

$$(4.4) \quad \gamma \varepsilon (\delta w_h, v_h)_h - \gamma \varepsilon (\nabla \delta u_h, \nabla v_h) - \frac{1}{\nu} (\chi_{\mathcal{A}(u_h)}^h \delta u_h, v_h)_h = -F_{\nu, h}^{(2)}(u_h, w_h)$$

for all  $v_h \in S_h$ , where  $\chi_{\mathcal{A}(u_h)}^h := \sum_{i=1}^N \chi_{\mathcal{A}(u_h)}^h(p_i) \varphi_i$  with  $\chi_{\mathcal{A}(u_h)}^h(p_i) = 0$  if  $0 \leq u_h(p_i) \leq 1$  and  $\chi_{\mathcal{A}(u_h)}^h(p_i) = 1$  otherwise. In matrix form, the linear system (4.3)–(4.4) reads as

$$(4.5) \quad \begin{bmatrix} M & -\gamma \varepsilon K - \frac{1}{\nu} G_A M G_A \\ K & (\frac{1}{\tau} + C_2) M + C_1 K \end{bmatrix} \begin{bmatrix} w^{(k+1)} \\ u^{(k+1)} \end{bmatrix} \\ = \begin{bmatrix} \frac{1}{\varepsilon} M \psi'_0(u^{\text{old}}) - \frac{1}{\nu} G_{A+} M G_{A+} \bar{u} \\ C_1 K u^{\text{old}} + (\frac{1}{\tau} + C_2) M u^{\text{old}} + \omega_0 G_D M G_D (f - u^{\text{old}}) \end{bmatrix},$$

where  $u^{(k+1)}, w^{(k+1)}, u^{\text{old}}, \bar{u} = (1, \dots, 1)^T \in \mathbb{R}^N$  ( $u^{\text{old}}$  is the solution from the previous time step) and for  $i = 1, \dots, N$

$$G_A = G_A(u^{(k)}) = \text{diag} \begin{pmatrix} 1 & \text{if } u^{(k)}(p_i) < 0 \text{ or } u^{(k)}(p_i) > 1 \\ 0 & \text{otherwise} \end{pmatrix},$$

$$G_{A+} = G_{A+}(u^{(k)}) = \text{diag} \begin{pmatrix} 1 & \text{if } u^{(k)}(p_i) > 1 \\ 0 & \text{otherwise} \end{pmatrix}$$

describe the matrix representations of the generalized derivatives. The lumped mass matrix and the stiffness matrix are defined as

$$M_{i,j} := (\varphi_i, \varphi_j)_h \quad \text{and} \quad K_{i,j} := (\nabla \varphi_i, \nabla \varphi_j) \quad \forall i, j = 1, \dots, N,$$

respectively, and

$$G_D = \text{diag} \begin{pmatrix} 1 & \text{if } x \in \Omega \setminus D \\ 0 & \text{if } x \in D \end{pmatrix}.$$

Note that  $M$  is a diagonal symmetric positive definite matrix and  $K$  is symmetric and positive semidefinite.

*Remark 4.1.* Using the smooth double-well potential, the linear system reads as

$$(4.6) \quad \begin{bmatrix} M & \gamma \varepsilon K \\ \gamma \varepsilon K & -\gamma \varepsilon ((\frac{1}{\tau} + C_2) M + C_1 K) \end{bmatrix} \begin{bmatrix} w \\ -u \end{bmatrix} \\ = \begin{bmatrix} \frac{1}{\varepsilon} M \psi'_s(u^{\text{old}}) \\ \gamma \varepsilon (C_1 K u^{\text{old}} + (\frac{1}{\tau} + C_2) M u^{\text{old}} + \omega_0 G_D M G_D (f - u^{\text{old}})) \end{bmatrix}.$$

**5. Preconditioning for the smooth modified Cahn–Hilliard inpainting.** For the system matrix in (4.6)

$$(5.1) \quad A = \begin{bmatrix} M & \gamma\varepsilon K \\ \gamma\varepsilon K & -\gamma\varepsilon\left(\frac{1}{\tau} + C_2\right)M + C_1K \end{bmatrix},$$

we propose the block-triangular preconditioner

$$P = \begin{bmatrix} M & 0 \\ \gamma\varepsilon K & -S \end{bmatrix},$$

motivated by [25]. Here,  $S$  is the Schur complement  $-\gamma\varepsilon\left(\frac{1}{\tau} + C_2\right)M + C_1K - \gamma^2\varepsilon^2KM^{-1}K$ . Preconditioners of this form have proven to perform well based on the simple observation that the preconditioned matrix  $P^{-1}A$  has a small number of distinct eigenvalues [44]. It is obvious that we never want to form the Schur complement explicitly, as the storage requirements for realistic scenarios would not be feasible. In order to represent most of the terms in  $S$ , we suggest the following approximation:

$$\begin{aligned} \hat{S} &= -\hat{S}_1M^{-1}\hat{S}_1 \\ &= -\left(\sqrt{\gamma\varepsilon\left(C_2 + \frac{1}{\tau}\right)M + \gamma\varepsilon K}\right)M^{-1}\left(\sqrt{\gamma\varepsilon\left(C_2 + \frac{1}{\tau}\right)M + \gamma\varepsilon K}\right), \end{aligned}$$

motivated by [48], where we use an algebraic multigrid (AMG) preconditioner for the approximation of the inverse of  $\hat{S}_1$ . AMG methods typically exhibit geometric-like properties for positive definite elliptic-type operators such as  $\hat{S}_1$  but use only algebraic information. This has the advantage that AMG can work well even for complicated geometries and meshes. We refer the reader to [52, 29] for more information on AMG. We also want to emphasize that geometric multigrid (GMG) (see, e.g., [61, 35]) approximations are also well suited to approximate  $\hat{S}_1$  provided they can be readily applied. First of all, our choice of  $\hat{S}$  fulfills the condition  $\hat{S} = S$ , if  $C_1 = 2\sqrt{\gamma\varepsilon\left(C_2 + \frac{1}{\tau}\right)}$ , which involves that our preconditioned matrix  $P^{-1}A$  has only two distinct eigenvalues in this case. Otherwise, the suitability of  $\hat{S}$  as a preconditioner for  $S$  is guaranteed by the following lemma.

**Lemma 5.1.** *The eigenvalues of  $\hat{S}^{-1}S$  are contained within the following interval:*

$$\lambda(\hat{S}^{-1}S) \in \left[ \frac{1}{2}, 1 + \frac{C_1}{2\sqrt{\gamma\varepsilon\left(C_2 + \frac{1}{\tau}\right)}} \right].$$

*Proof.* As both  $S$  and  $\hat{S}$  are symmetric matrices, we may prove the result using a Rayleigh quotient argument. We write that

$$\begin{aligned} \frac{v^T S v}{v^T \hat{S} v} &= \frac{\gamma^2 \varepsilon^2 v^T K M^{-1} K v + \gamma \varepsilon (C_2 + \frac{1}{\tau}) v^T M v + \gamma \varepsilon C_1 v^T K v}{\gamma^2 \varepsilon^2 v^T K M^{-1} K v + \gamma \varepsilon (C_2 + \frac{1}{\tau}) v^T M v + 2\gamma \varepsilon \sqrt{\gamma \varepsilon (C_2 + \frac{1}{\tau})} v^T K v} \\ &= \frac{1 + \frac{\gamma \varepsilon C_1 v^T K v}{\gamma^2 \varepsilon^2 v^T K M^{-1} K v + \gamma \varepsilon (C_2 + \frac{1}{\tau}) v^T M v}}{1 + \frac{2\gamma \varepsilon \sqrt{\gamma \varepsilon (C_2 + \frac{1}{\tau})} v^T K v}{\gamma^2 \varepsilon^2 v^T K M^{-1} K v + \gamma \varepsilon (C_2 + \frac{1}{\tau}) v^T M v}}. \end{aligned}$$

The quantity  $\frac{2\gamma \varepsilon \sqrt{\gamma \varepsilon (C_2 + \frac{1}{\tau})} v^T K v}{\gamma^2 \varepsilon^2 v^T K M^{-1} K v + \gamma \varepsilon (C_2 + \frac{1}{\tau}) v^T M v}$  arising in the denominator may be written as  $\frac{2a^T b}{a^T a + b^T b}$ ,  $a = \sqrt{\gamma \varepsilon (C_2 + \frac{1}{\tau})} M^{\frac{1}{2}} v$  and  $b = \gamma \varepsilon M^{-\frac{1}{2}} K v$ . Now, as in the work of Pearson et al. (see [46, 49, 47]), since  $a^T a > 0$  (because of the positive definiteness of  $M$ ), we may use that  $(a - b)^T (a - b)$  is nonnegative to bound this quantity above by one. On the other hand, the quantity is clearly bounded below by zero. The quantity  $\frac{\gamma \varepsilon C_1 v^T K v}{\gamma^2 \varepsilon^2 v^T K M^{-1} K v + \gamma \varepsilon (C_2 + \frac{1}{\tau}) v^T M v}$  arising in the numerator may be written as  $\frac{2C_1 a^T b}{2\sqrt{\gamma \varepsilon (C_2 + \frac{1}{\tau})} (a^T a + b^T b)}$ , so it is bounded above by  $\frac{C_1}{2\sqrt{\gamma \varepsilon (C_2 + \frac{1}{\tau})}}$  and below by zero. Putting the work together, we have proved that

$$\lambda(\hat{S}^{-1} S) \in \left[ \frac{1}{2}, 1 + \frac{C_1}{2\sqrt{\gamma \varepsilon (C_2 + \frac{1}{\tau})}} \right]. \quad \blacksquare$$

Analyzing the result of Lemma 5.1, we first have that the eigenvalues are clustered away from the origin at 0.5 independent of any parameter. The upper bound includes a benign dependency between  $C_1$ ,  $\varepsilon$ , and  $C_2$  (as we work with  $\gamma = 1$  and  $\tau = 1$ , both can be neglected). Together with the bounds  $C_1 > \frac{1}{\varepsilon}$  and  $C_2 > \omega_0$  from the convexity splitting, the upper bound in Lemma 5.1 fulfills  $1 + \frac{C_1}{2\sqrt{\gamma \varepsilon (C_2 + \frac{1}{\tau})}} < 1 + \frac{1}{2} \sqrt{\frac{C_1^3}{\omega_0}}$ . Remember,  $\omega_0$  is the very large fidelity constant that keeps the solution close to the given image. The parameter  $C_1$  can be any value greater than  $\frac{1}{\varepsilon}$ . Adjusting  $C_1$  can achieve nice upper bounds in Lemma 5.1.

**6. Preconditioning for nonsmooth modified Cahn–Hilliard inpainting.** For the system matrix in (4.5), which we write as

$$(6.1) \quad A = \begin{bmatrix} M & -L \\ K & (\frac{1}{\tau} + C_2)M + C_1 K \end{bmatrix},$$

we propose as in the previous section the block-triangular preconditioner

$$P = \begin{bmatrix} M & 0 \\ K & -S \end{bmatrix}.$$

Here,  $S$  is the Schur complement  $(\frac{1}{\tau} + C_2)M + C_1K + KM^{-1}L$ . For  $S$  we suggest the following approximation:

$$\hat{S} = \hat{S}_2 M^{-1} \hat{S}_3 = \left( \sqrt{\frac{1}{\tau} + C_2 M + K} \right) M^{-1} \left( \sqrt{\frac{1}{\tau} + C_2 M + L} \right),$$

where we use as in the previous section an AMG preconditioner for the approximation of the inverse of  $\hat{S}_2$  and  $\hat{S}_3$ . Because we are dealing with nonsymmetry here, it is hard to prove eigenvalue bounds as in the symmetric case from the previous section. In the following, we want to illustrate the performance of  $\hat{S}^{-1}S$ . We therefore consider the eigenvalue problem

$$Sv = \lambda \hat{S}v$$

via MATLAB and analyze the robustness of the Schur complement approximation with respect to the penalty parameter  $\nu$  and the mesh parameter  $h$ . For the former, we vary  $\nu$  by the sequence  $\nu \in \{10^{-1}, 10^{-3}, 10^{-5}, 10^{-7}\}$ , and for the latter, we consider  $h \in \{2^{-4}, 2^{-5}, 2^{-6}\}$ . Moreover, we choose  $\tau = 1$ ,  $\gamma = 1$ ,  $C_1 > \frac{1}{\varepsilon}$ , and  $C_2 > \omega_0$ . In the following we present some eigenvalue distributions for different values of  $\varepsilon$ .

We start with the rather large choice  $\varepsilon = 0.8$ , which is typical when having a large inpainting gap; see Figure 1(a). For this case, we obtain only real eigenvalues and a few clusters whose number stays almost constant for varying  $\nu$  and  $h$ . Increasing the parameter  $C_2$  leads to a better clustering because the eigenvalues move closer together; see Figure 1(b). Next, we decrease  $\varepsilon$  to 0.1 in Figure 1(c) and get complex eigenvalues. The imaginary parts are clustered in a circle centered on 0.6 and of radius about 0.1 for varying  $\nu$  and  $h$ . The real parts are considered separately in Figure 1(d). Comparing to the eigenvalues in Figure 1(b), the values here look nearly the same but with a hint of a little improvement. Decreasing  $\varepsilon$  to 0.01 also leads to complex eigenvalues, which are shown in Figure 2(a), and again, considering the real parts in Figure 2(b), the clustering has improved. The imaginary parts are clustered in a circle centered on 0.7 and of radius about 0.2, and the real parts are mostly distributed about  $10^{-0.1}$  for varying  $\nu$  and about 1 for varying  $h$ . In the last example, we switch to  $\varepsilon = 0.001$ . Figure 2(c) shows that real eigenvalues occur around 1, 3, and 5 for varying values of  $\nu$ . For varying  $h$  we obtain complex eigenvalues whose imaginary parts are clustered like in the previous example. Considering the real parts in Figure 2(d), we again recognize the small number of clusters for varying  $h$ .

We have seen that we have obtained different eigenvalue distributions for several examples. However, we also observed only a small number of eigenvalue clusters, which justifies our choice of  $\hat{S}$ .

For the application of  $P$ , it remains to analyze the computational cost of solving a linear system with the block  $M$ . The aim of a preconditioner is to resemble the original matrix by also being easy to invert. As  $M$  is a diagonal matrix it is a simple task to apply the inversion.

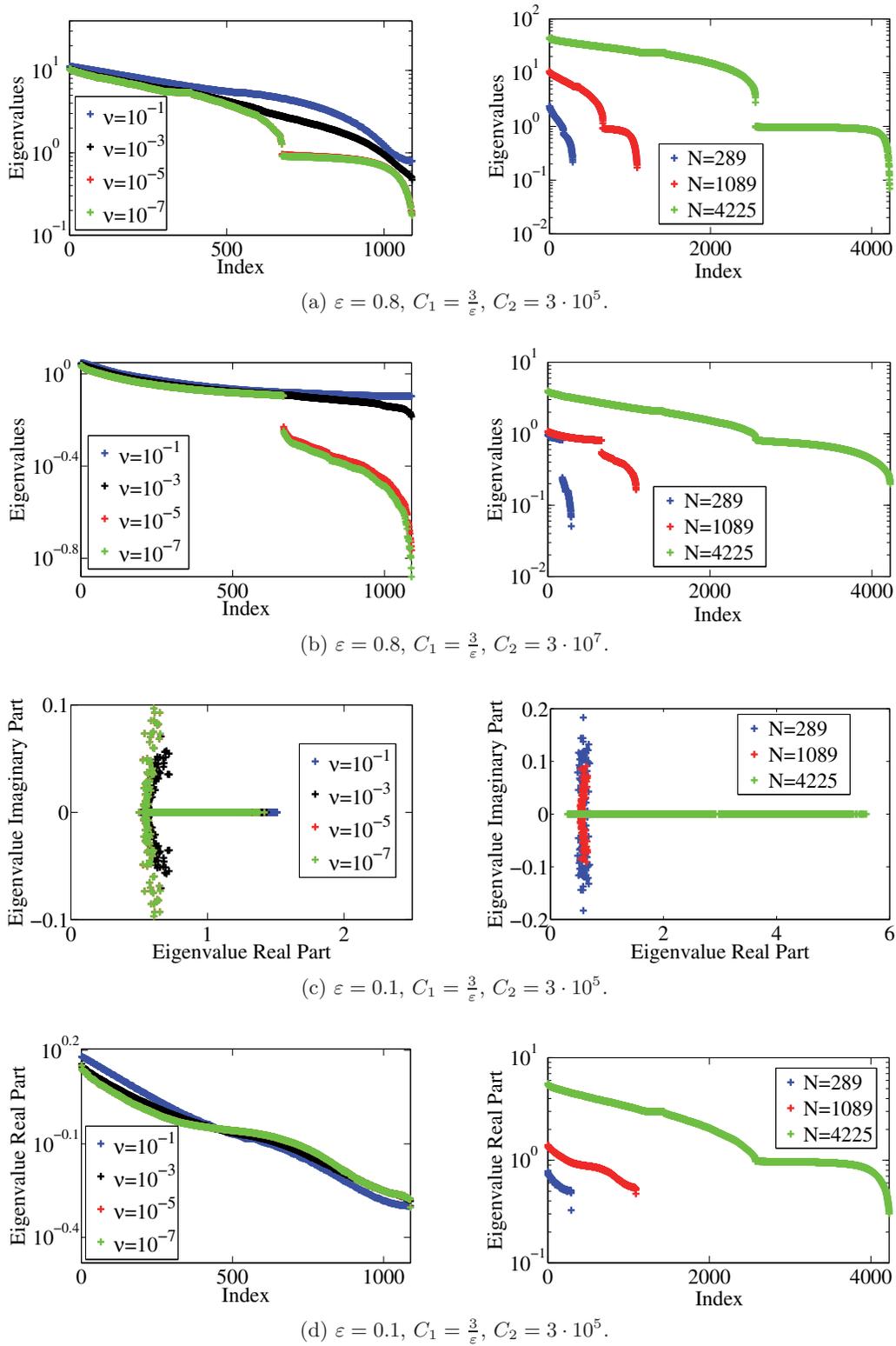
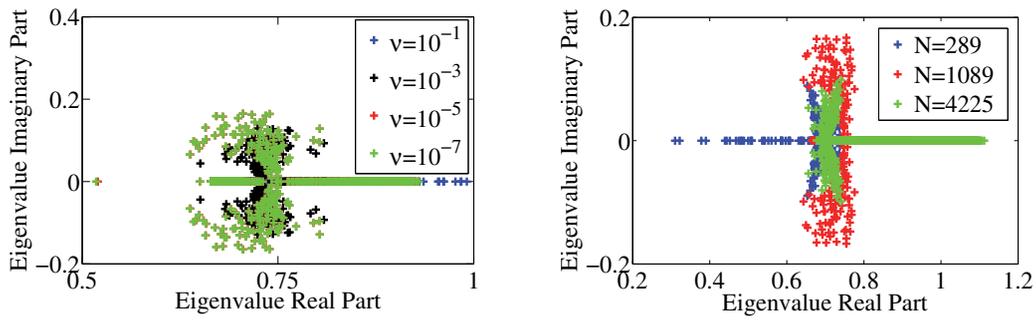
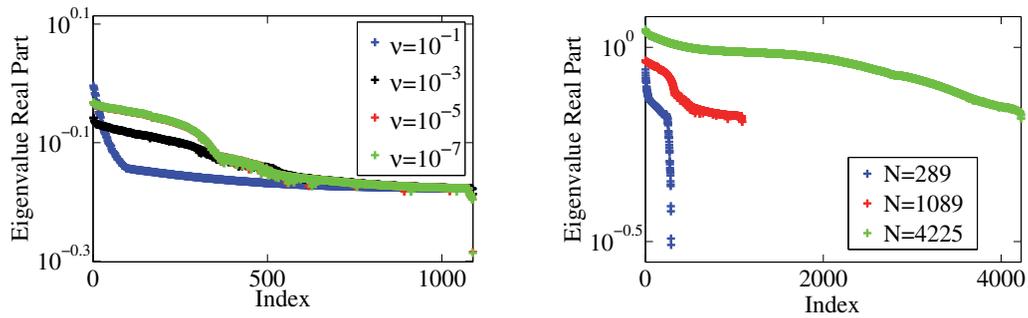


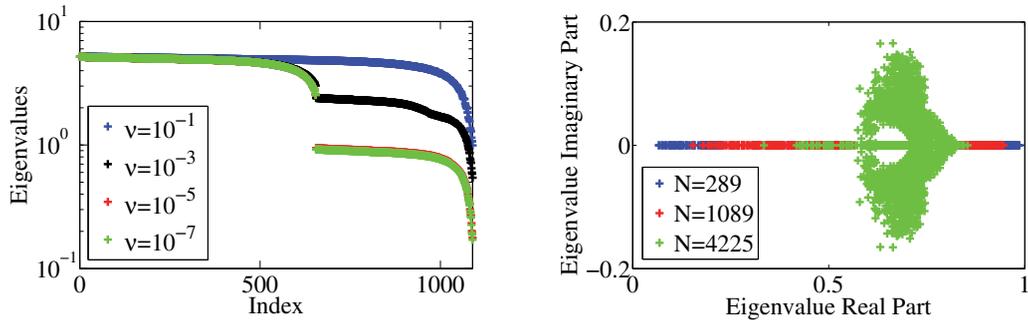
Figure 1. Eigenvalues for the Schur complement approximation.



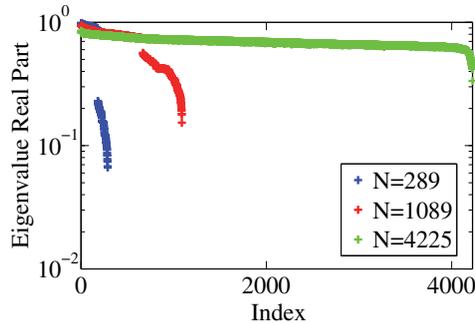
(a)  $\varepsilon = 0.01, C_1 = \frac{1}{\varepsilon}, C_2 = 3 \cdot 10^5$ .



(b)  $\varepsilon = 0.01, C_1 = \frac{1}{\varepsilon}, C_2 = 3 \cdot 10^5$ .



(c)  $\varepsilon = 0.001, C_1 = \frac{1}{\varepsilon}, C_2 = 3 \cdot 10^5$ .



(d)  $\varepsilon = 0.001, C_1 = \frac{1}{\varepsilon}, C_2 = 3 \cdot 10^5$ .

Figure 2. Eigenvalues for the Schur complement approximation.

**7. Numerical results.** In this section we show results for the modified Cahn–Hilliard equation. The preconditioners we present, whereby we employ left preconditioning, can be embedded into various Krylov subspace solvers. For the nonsymmetric matrix (6.1) we propose the use of a nonsymmetric short-term recurrence method, namely BiCG [30], but note that also other solvers such as QMR [31], BiCGSTAB [59], or GMRES [54] can be used with this preconditioner. For the numerical results presented in this section we set the BiCG tolerance to be  $10^{-7}$  for the preconditioned relative residual in all examples. For the multilevel approximations we choose Trilinos AMG approximations [36]. For one application of the preconditioner we take 10 steps of a Chebyshev smoother and two V-cycles. The discretization is performed with the finite element package deal.II [1, 2], which allows the use of the Trilinos library. All numerical experiments listed here are generated with finite elements on rectangles. For the SSN method we use the stopping criterion in [38, 10], given by

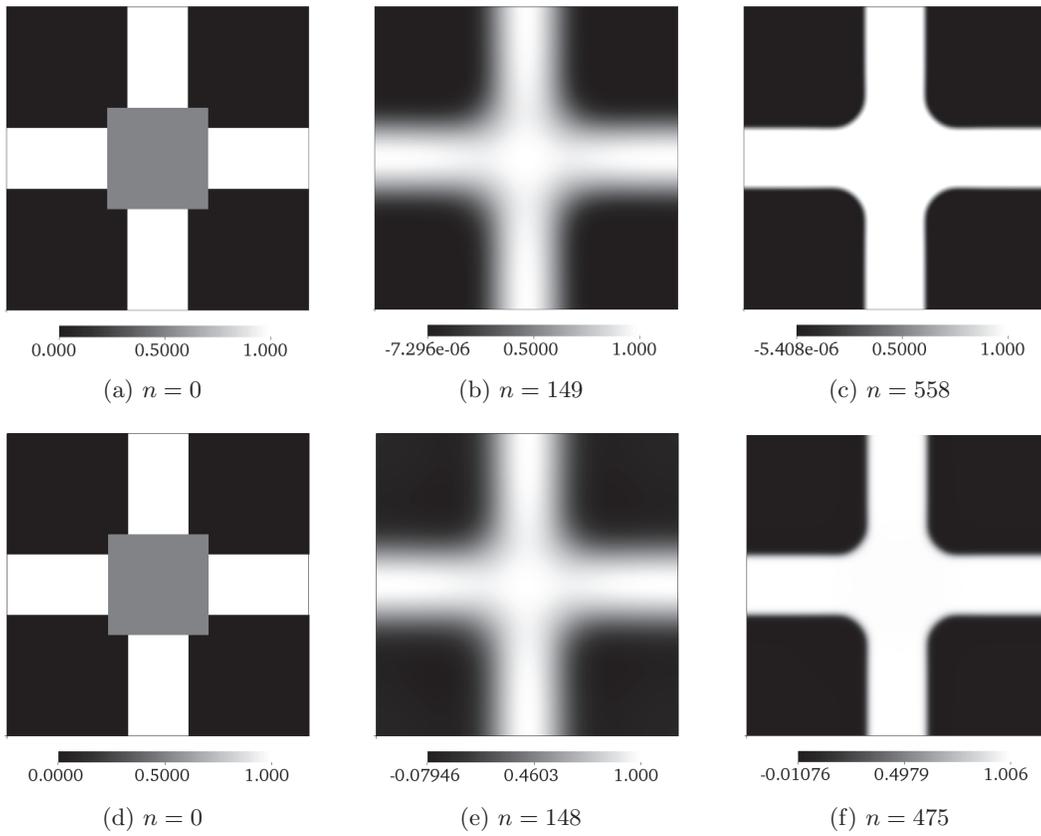
$$\|F_\nu(u_h^{(k)}, w_h^{(k)})\|_2 \leq \epsilon_{\text{rel}} \|F_\nu(u_h^{(0)}, w_h^{(0)})\|_2 + \epsilon_{\text{abs}}, \quad k = 1, \dots, k_{\text{max}},$$

where we set  $k_{\text{max}} = 100$ ,  $\epsilon_{\text{rel}} = 10^{-12}$ , and  $\epsilon_{\text{abs}} = 10^{-6}$  in all examples. For the handling of the parameter  $\nu$ , we follow [38, 10] and solve (4.1)–(4.2) for the sequence  $\nu_1 = 10^{-1} \geq \nu_2 = 10^{-2} \geq \dots \geq \nu_{\text{max}}$ , where we initialize each Newton method by the approximate solution of the previously used  $\nu$  value. After a few time steps, we fix  $\nu = \nu_{\text{max}}$ . This is because the initial solutions at the beginning might not be a good starting point for the SSN methods. In all examples, we denote the inpainting region in gray. Moreover, we use the two-scale approach described in [33, 6] which is successful in connecting edges across large inpainting regions. There, the first simulation is run close to steady state for a large choice of  $\varepsilon$ . We use as stopping criterion

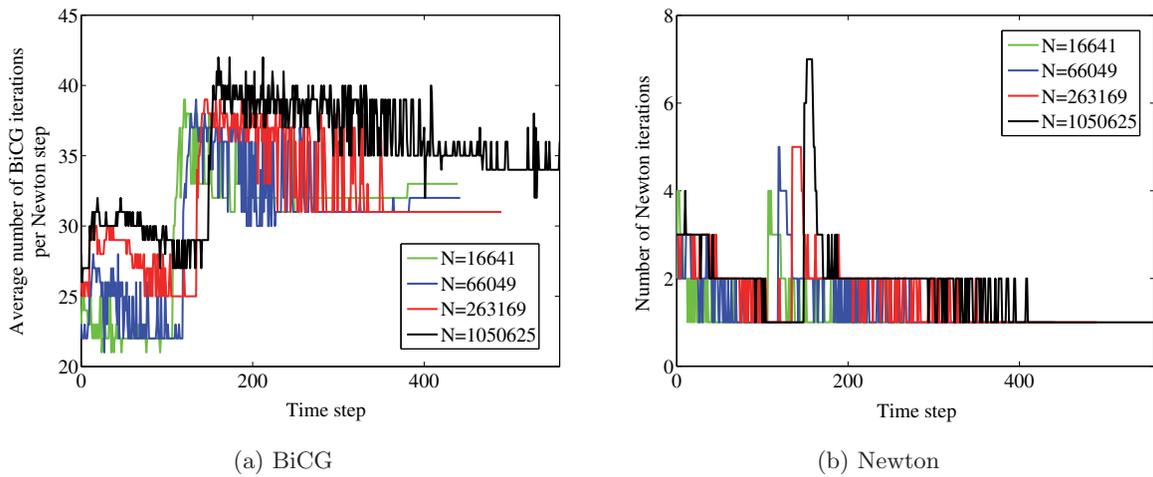
$$\|u_h^{(n)} - u_h^{(n-1)}\|_2 \leq \epsilon,$$

where we set  $\epsilon = 10^{-2}$ . Then, the obtained approximate solution is used as initial state for the second run with a small  $\varepsilon$  to sharpen the edges. For this, we use the same stopping criterion as before, but with  $\epsilon = 10^{-4}$ . Moreover, we set  $\gamma = 1$ ,  $\tau = 1$ , and  $C_2 = 3\omega_0$  in all examples ( $\omega_0$  defined in (2.5) is the large parameter making sure that the result is close enough to the original picture).

**7.1. Iteration numbers.** We want to look at the number of BiCG iterations as well as the number of Newton iterations (in the nonsmooth case) for different uniform mesh sizes. The test example consists of a cross that contains the damaged area in the middle. The corresponding simulation is shown in Figure 3 (here  $h = 2^{-10}$ ) for the nonsmooth (above) and smooth (below) Cahn–Hilliard inpainting models. Here and in the following,  $n$  denotes the time step, as introduced in the convexity splitting scheme. In order to show the influence of the two-scale approach ( $\varepsilon$ -jump), as mentioned in the beginning of section 7, we apply this method here. At first, the modified Cahn–Hilliard equations are run for the large interface parameter  $\varepsilon = 0.8$ . Figures 3(b) and 3(e) show the successful connection of the edges. Then, we switch to  $\varepsilon = 0.01$  to sharpen the edges. The results of both models are given in Figures 3(c) and 3(f). The parameters are set as  $\omega_0 = 100000$  and  $C_1 = 300$ .



**Figure 3.** Nonsmooth (above) and smooth (below) modified Cahn–Hilliard evolution for the cross computation.



**Figure 4.** Iteration numbers for the nonsmooth model.

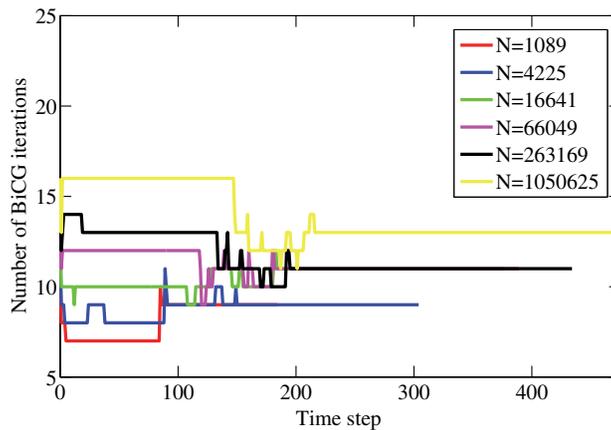


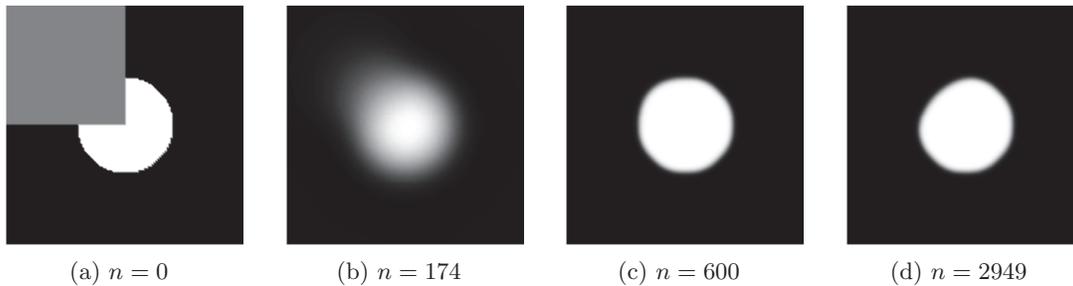
Figure 5. Iteration numbers for the smooth model.

Figure 4 shows the number of BiCG iterations per Newton step and the number of Newton iterations per time step for the nonsmooth model. In both figures, the number of unknowns  $N$  is listed. We observe a little growth after the  $\varepsilon$ -jump. For the SSN, we use the sequence with  $\nu_{\max} = 10^{-7}$  for the first 10 time steps. We can reduce the first few Newton iteration numbers after the  $\varepsilon$ -jump if we also use the  $\nu$ -sequence here for some time steps. For one Newton step during the evolution with  $\varepsilon = 0.8$ , the computational times (in seconds) on a Linux desktop system using a Core 2 processor are 0.15, 0.56, 2.75, 18.75, 90.75, 355.41 for mesh sizes  $h = 2^{-5}, \dots, 2^{-10}$ , respectively. The BiCG iteration numbers for the smooth model are illustrated in Figure 5. We have only a benign increase for the coarser grids. The corresponding computational times for one time step (in seconds) are 0.07, 0.32, 1.45, 9.01, 40.88, 202.14 for mesh sizes  $h = 2^{-5}, \dots, 2^{-10}$ , respectively. In fact, we recognize a decrease of iteration numbers for the finer grids. All in all, the numbers of BiCG and Newton iterations stay low and we observe nearly mesh independent iteration numbers.

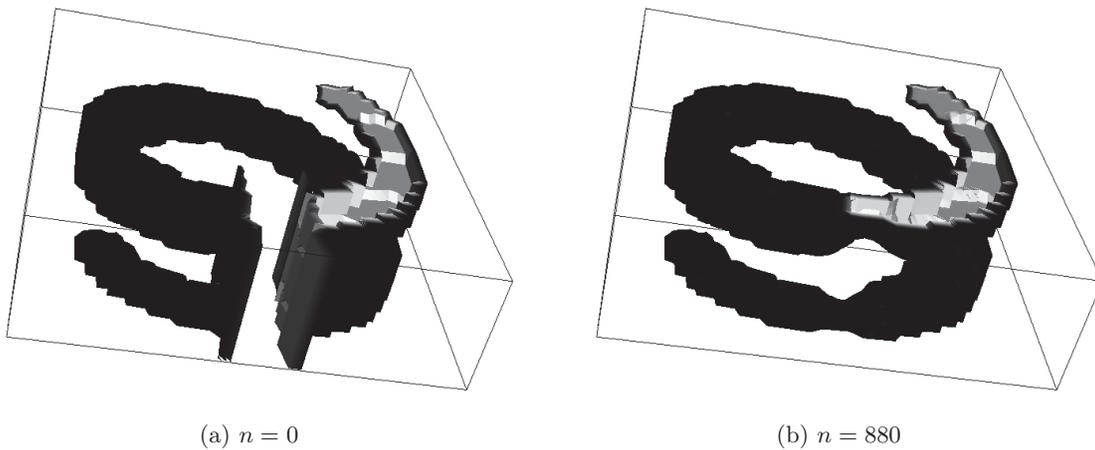
**7.2. Comparison of the use of finite elements and Fourier transforms.** Solving ODEs or PDEs via spectral methods is an important tool in image processing. As the FFT basis functions are eigenvectors of the difference operators, which form the discrete Laplacian, FFT methods rapidly solve divergence equations on simple domains.

For the smooth modified Cahn–Hilliard equation using the double-well potential, Bertozzi et al. (see [33, 6, 12]) proposed a two-dimensional FFT method and achieved fast inpainting; see Figure 6. The final result here is obtained after only 4524.07 seconds, where  $h = 2^{-7}$  and all the other parameters are set as in the previous section. The figure also shows the first disadvantage in the use of a smooth potential. Compared to the end result (which we have obtained due to the convergence tolerance given at the beginning of section 7), the solution after  $n = 600$  time steps gives the rounder (and better) circle. This means that first we should stop earlier in this example, in order to get the better visual inpainted picture. But this, in fact, includes the questions of when we should stop and how to check this. The second fact is that the steady state solution is not exactly the result that we would expect. The same observations are made when using FEM; see section 7.4. In fact, concerning the computational time, this method is hard to beat, but with respect to more complex inpainting-type problems,

spectral methods on complicated domains are difficult; see, e.g., [11]. Figure 7 shows a 3D example on a cube with a cylindrical hole inside (see the mesh in Figure 8) to which our method can be applied without problems. The usage of finite elements typically allows us to compute missing information on arbitrary domains.

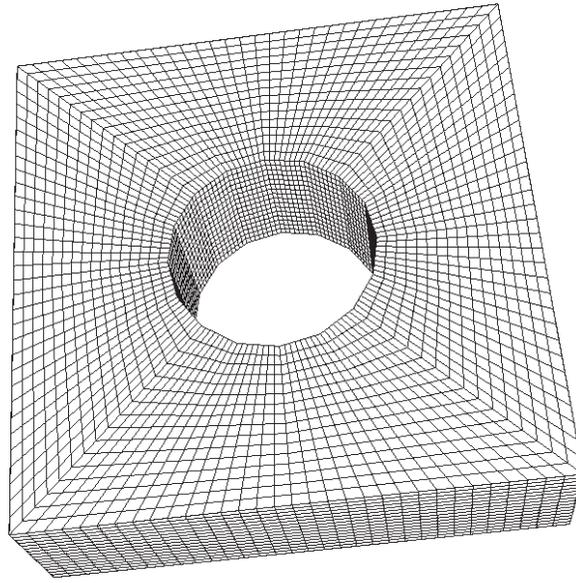


**Figure 6.** Smooth modified Cahn–Hilliard evolution for the circle computation with FFT.

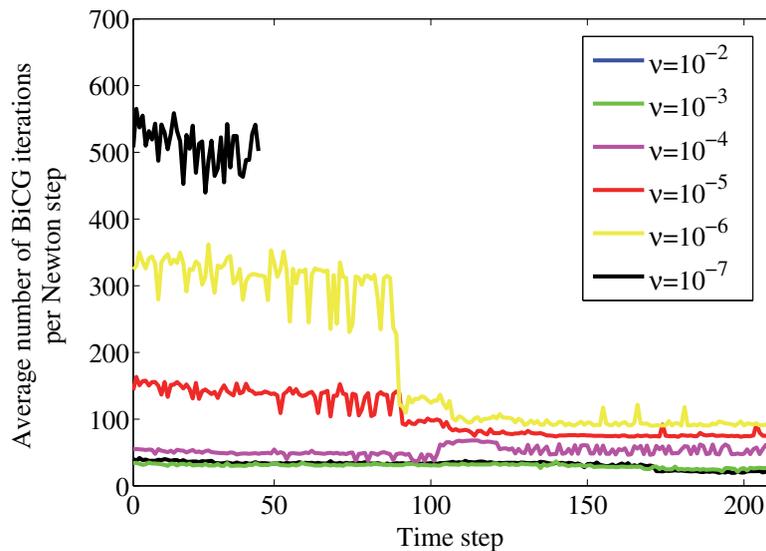


**Figure 7.** Smooth modified Cahn–Hilliard evolution for a 3D spiral helix on a complicated domain.

Extending the idea of using an FFT-based solution scheme to the nonsmooth problem is a challenge. An efficient FFT-based implementation employing the nonsmooth potential would typically suffer from the nonconstant coefficient matrix coming from the term  $G_A M G_A$  that originates in the discretization of the max- and min-terms; see Eyre [28] for an application with a smooth potential for the standard Cahn–Hilliard equation. This poses a challenge, as the optimal approximation of the nonconstant term needs to be of good quality to achieve small iteration numbers of the linear solver. In general, it holds that for spectral methods, the smoother the function, the faster the convergence. Figure 9 shows how the iteration numbers increase with the nonsmoothness obtained by varying the penalty parameter  $\nu$ . The decrease of iteration numbers after around 100 time steps is a result of the  $\varepsilon$ -jump.



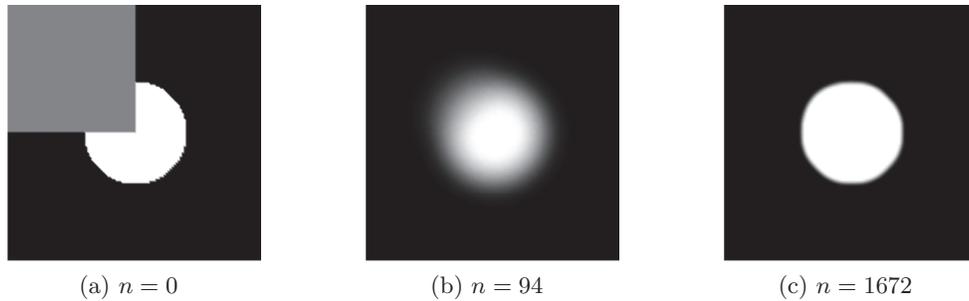
**Figure 8.** Mesh of a cube with a cylindrical hole.



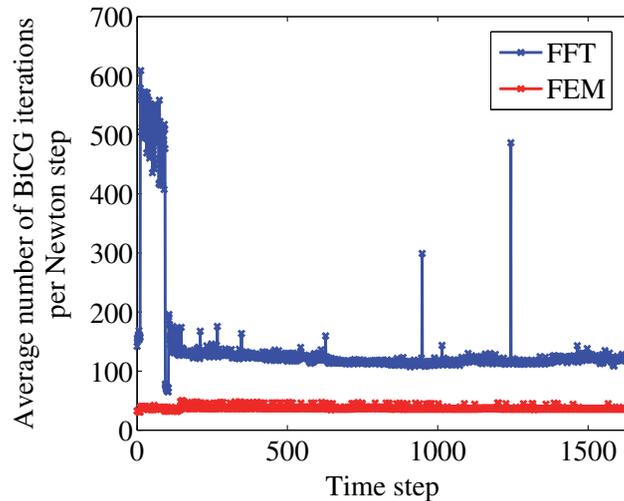
**Figure 9.** Iteration numbers for the nonsmooth model using FFT for the application of the preconditioner.

The visual example for the circle computation is given in Figure 10. The final result here is obtained after 4775.26 seconds. Compared to the results obtained by the smooth model before (see Figure 6), the nonsmooth model gives the better visual pictures; see also section 7.4 for more details. The iteration numbers for this example are shown in Figure 11, together with the iteration numbers, when using FEM instead. It again illustrates the

difficulty of nonsmoothness in FFT-based methods. This is another motivation to focus on a discretization via finite elements for the nonsmooth potential function.



**Figure 10.** Nonsmooth modified Cahn–Hilliard evolution for the circle computation with FFT.



**Figure 11.** Iteration numbers for the nonsmooth model.

**7.3. Handling of convexity splitting.** This section confirms the argument made in Remark 2.2, that keeping the convexity term  $C_1$  within the nonsmooth Cahn–Hilliard inpainting model gives better preconditioning results. Omitting the term  $C_1$  leads to the Schur complement

$$S_{wC_1} = \left( \frac{1}{\tau} + C_2 \right) M + KM^{-1}L,$$

which has one less term than the Schur complement  $S = (\frac{1}{\tau} + C_2)M + C_1K + KM^{-1}L$  we are working with in (6.1). This means that the Schur complement preconditioner of the form

$$\hat{S} = \left( \sqrt{\frac{1}{\tau} + C_2M + K} \right) M^{-1} \left( \sqrt{\frac{1}{\tau} + C_2M + L} \right)$$

is a better approximation for the Schur complement  $S$  in (6.1) than for  $S_{wC_1}$ . Iteration numbers, obtained by performing inpainting of a zebra (as in section 7.6), illustrate our observations; see Figure 12. The parameters are  $h = 2^{-8}$ ,  $\omega_0 = 10^6$ ,  $C_1 = 1000$ , and  $\varepsilon = 0.8 \rightarrow 0.01$ .

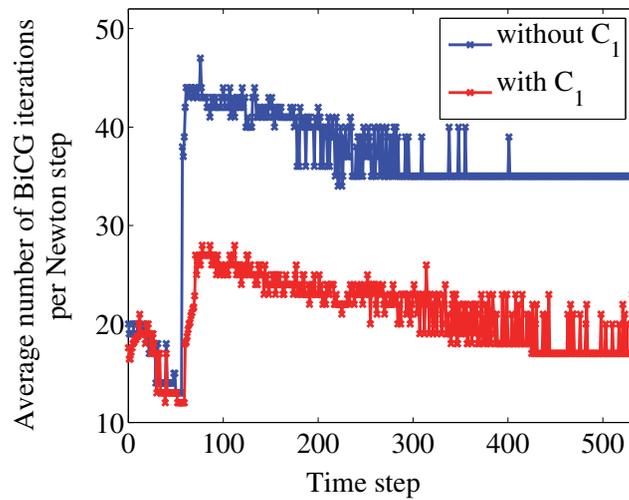
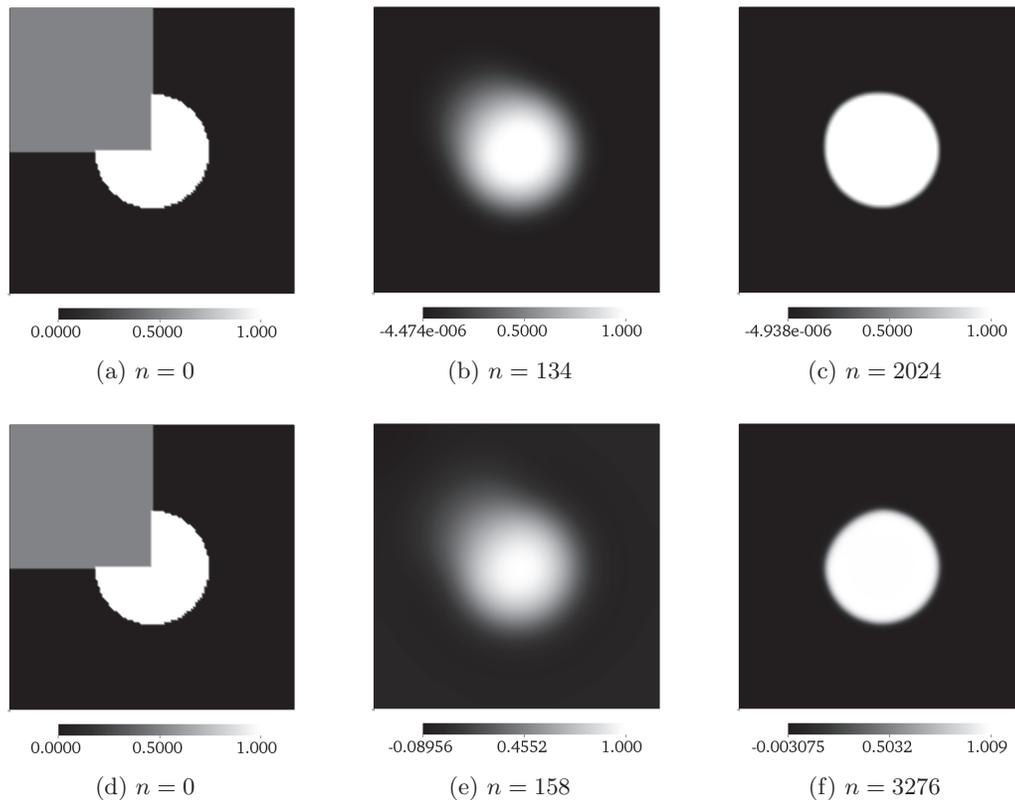


Figure 12. Iteration numbers for two different convexity splitting schemes.

In summary, keeping the convexity term  $C_1$  gives better iteration numbers and the additional costs are limited.

**7.4. Comparison of the smooth and nonsmooth model.** We want to compare the performance of the smooth and nonsmooth Cahn–Hilliard inpainting models (using FEM) on the damaged circle in Figure 13. Because of the large inpainting domain, we use the two-scale approach with  $h = 2^{-7}$  and the same choice for all the other parameters as in section 7.1. Figures 13(b) and 13(e) show the successful connection of the edges as intermediate results. Then, the final outputs are given in Figures 13(c) and 13(f) after the  $\varepsilon$ -jump. This is a well-suited example because of the smoothing property of the Cahn–Hilliard equation. Comparing both results, the rounder circle as well as the more intense colors (which can be seen by comparing the minimum and maximum values of the colors) are obtained in the nonsmooth model. This verifies our preference for using a nonsmooth potential.



**Figure 13.** *Non-smooth (above) and smooth (below) modified Cahn–Hilliard evolutions for the circle computation with FEM.*

**7.5. Comparison to other methods.** The circle example is used to show visual comparisons to other inpainting models. The methods we test against are the MATLAB function called `inpaintn` (MATLAB) as well as three other higher-order TV-based inpainting models provided by Schönlieb (see [55, 12]), which are inpainting methods using the heat equation (heat), standard TV inpainting (TV), and a fourth-order TV PDE for inpainting (TV4). All the MATLAB files are available on the Web.<sup>1</sup> The examples are performed on a Linux desktop system using a Core 2 processor and run with MATLAB for 10000 time steps. Figure 14 shows the results of each inpainting method. Additionally, we have added the results obtained by the smooth/non-smooth Cahn–Hilliard inpainting models using FEM/FFT that we have already seen in this section. Comparing the visual results, the Cahn–Hilliard inpainting method outperforms all the other approaches presented here. All simulations are run with standard parameters. Maybe one could obtain better results for the other approaches with a fine tuning of the parameters or after more iterations. But as our methods were already converged after less than 3500 time steps (see sections 7.2 and 7.4) we stopped the other methods. Concerning computational times, the other methods are at an advantage. For 10000 iterations, the times are (in seconds) 26.02, 190.84, 230.76, and 242.82 (for the upper row in Figure 14 from the left to the right). For the lower row, the computational times (in

<sup>1</sup><http://www.mathworks.se/matlabcentral/fileexchange/34356-higher-order-total-variation-inpainting>

seconds) needed to achieve our convergence tolerance (see the beginning of section 7) are from the left to the right 11902.25 (2024), 4657.17 (3276), 4775.26 (1672), 4524.07 (2949), where the number of time steps is given in brackets. Comparisons to some of the inpainting methods mentioned in section 1 can be found, e.g., in [33].

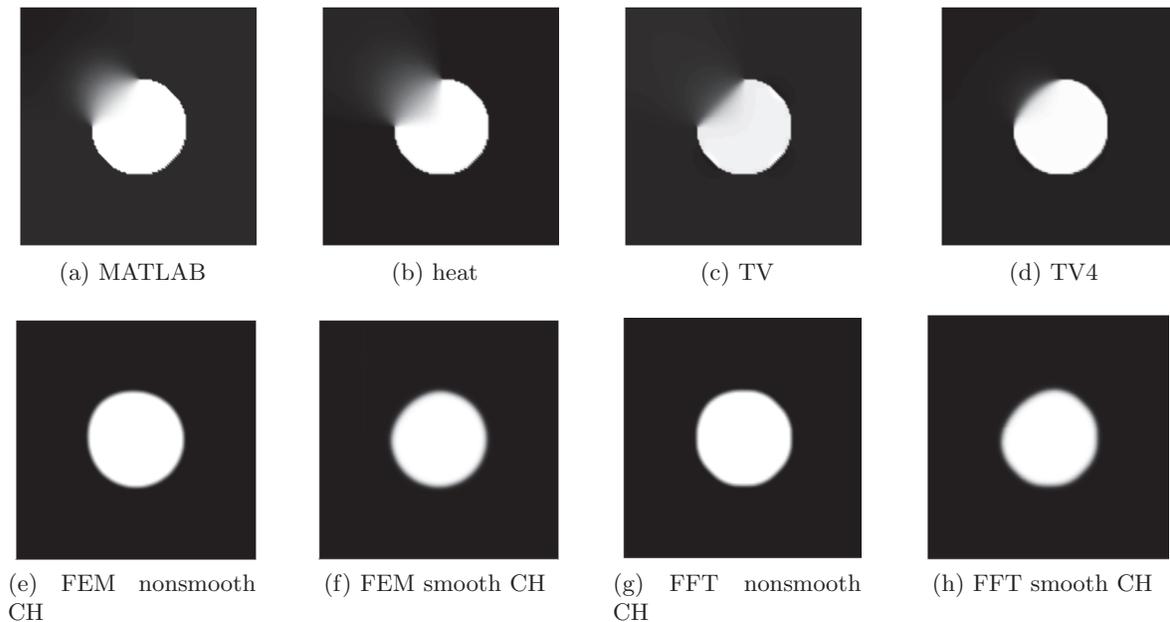
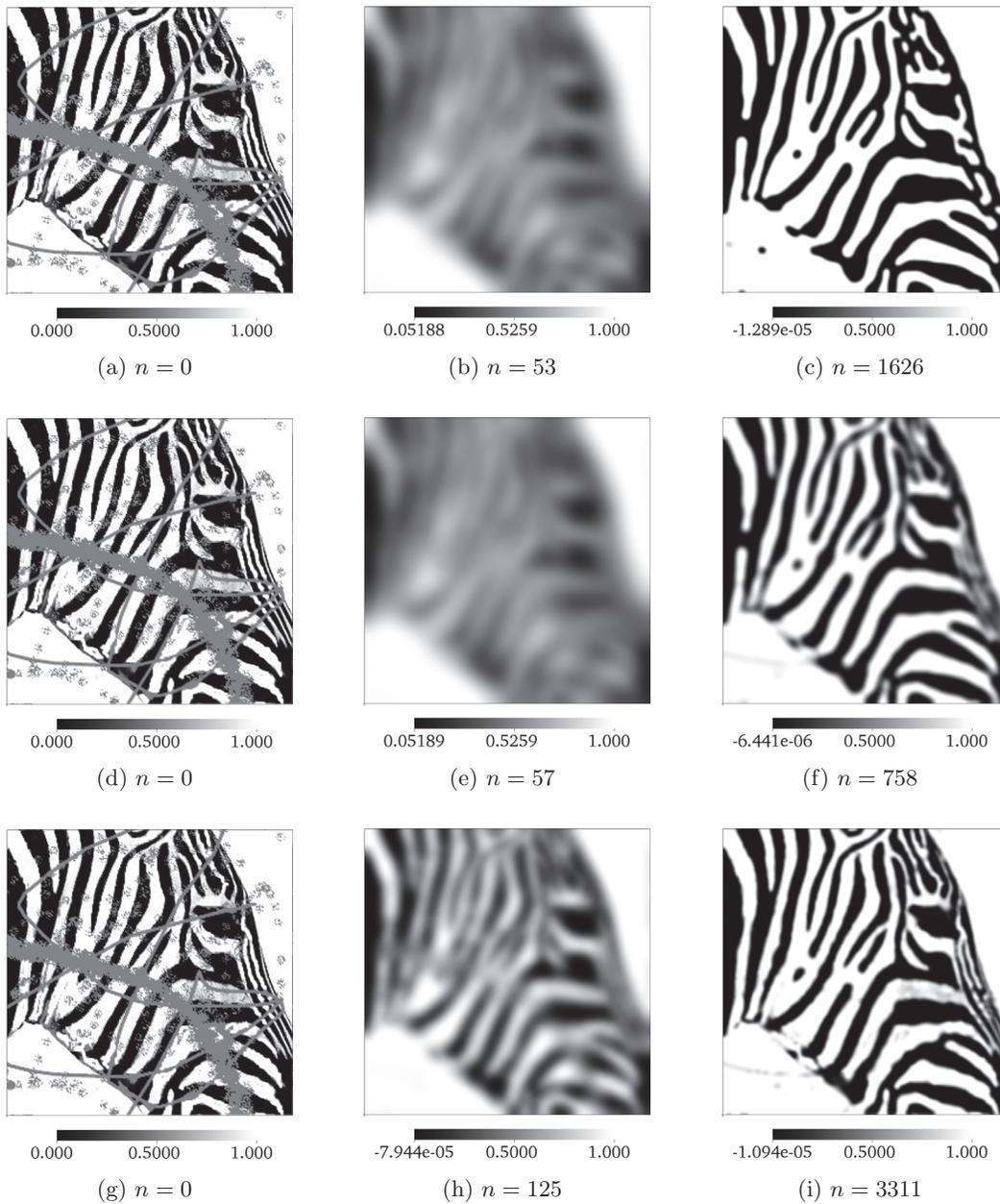


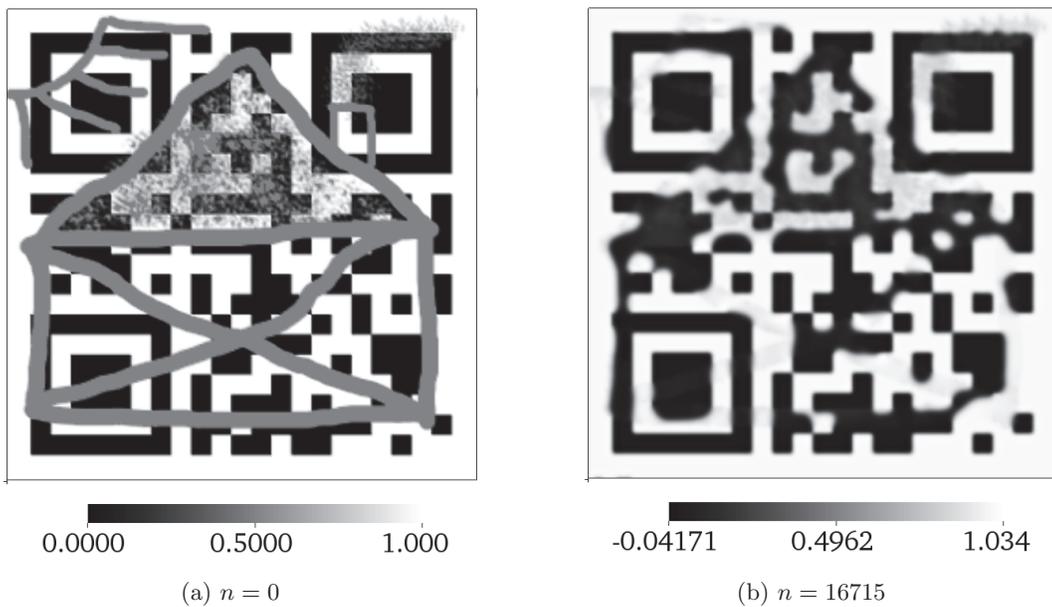
Figure 14. Results for the circle inpainting test.

**7.6. Zebra.** We now look at a real and more complex picture. For binary images, a very suitable example is a photograph of a zebra. Figure 15 shows how the nonsmooth modified Cahn–Hilliard model may be applied to the inpainting of an extract of a plains zebra photo for three different sets of parameters. The initial value for  $\varepsilon$  is 0.8 and we use  $h = 2^{-8}$  in all simulations. In the first example (first row in Figure 15), we execute an  $\varepsilon$ -jump to 0.005 and use  $C_1 = \frac{3}{\varepsilon}$  as well as  $\omega_0 = 10^6$ . We observed at most 50 BiCG iterations. The second row shows a simulation with an  $\varepsilon$ -jump to 0.01 as well as  $C_1 = 1000$  and  $\omega_0 = 10^6$ , where we obtained at most 35 BiCG iterations. In the last example, we switch to the  $\varepsilon$ -jump 0.007 and set  $C_1 = 7000$  and  $\omega_0 = 1.5 \cdot 10^7$  and gained at most 45 BiCG iterations. The three simulations show the influence of different parameter settings. We get the most intensive colors in the first example when  $\varepsilon$  and  $C_1$  are relatively small. Increasing  $\omega_0$  results in a more exact inpainting version in the area  $\Omega \setminus D$  in the sense that thin black/white areas are captured more accurately; see the border of the zebra below the eye in the last example. By means of time steps, it can be observed that the larger the  $C_1$ , the more time steps that are needed.



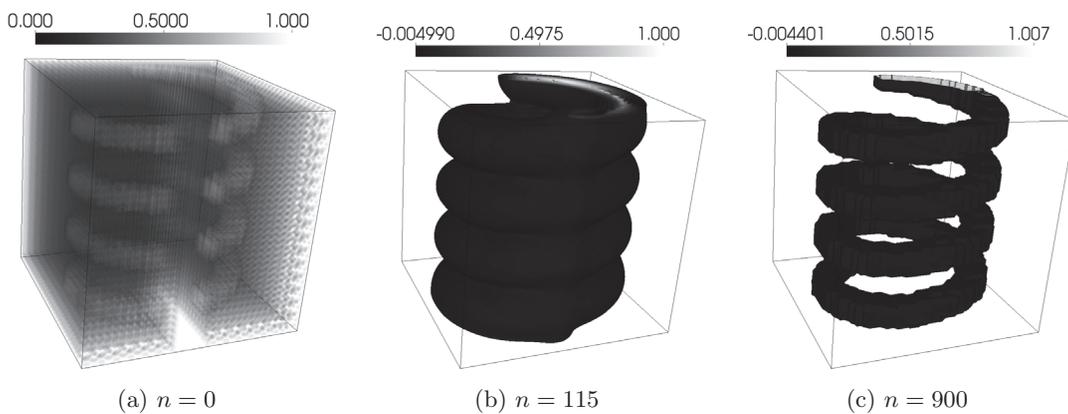
**Figure 15.** Nonsmooth modified Cahn–Hilliard evolution for the zebra computation. ©2012 Thomas Rolle from the Zoo Magdeburg, Germany. Used with permission.

**7.7. QR code.** Now, we test the inpainting model for a QR code. Clearly, this is a very difficult example for Cahn–Hilliard inpainting because of the rectangular structures. Nevertheless, the result is satisfying; see Figure 16. Here, we use  $\varepsilon = 0.005$  during the whole computation as well as  $C_1 = 2\sqrt{\gamma\varepsilon(\frac{1}{\tau} + C_2)}$ ,  $\omega_0 = 5 \cdot 10^7$ . This choice for  $C_1$  is optimal in the sense that we need only 2 BiCG iterations in every time step; see section 5.



**Figure 16.** Smooth modified Cahn–Hilliard evolution for the QR code computation.

**7.8. Inpainting in three dimensions.** Last, but not least, we perform inpainting in three dimensions on a damaged spiral helix contained in a noisy box; see Figure 17(a). We set  $\varepsilon = 2 \rightarrow 0.01$ ,  $h = 2^{-5}$ ,  $C_1 = 1000$ , and  $\omega_0 = 10^5$  and apply the nonsmooth Cahn–Hilliard model. We observe at most 60 BiCG iterations.



**Figure 17.** Nonsmooth modified Cahn–Hilliard evolution for 3D inpainting.

**8. Conclusions.** In this paper, we have shown the practicability of a nonsmooth potential for the modified Cahn–Hilliard equation. We have proposed a Moreau–Yosida regularization technique for the pointwise constraints and proved superlinear convergence of the SSN method in function space. Moreover, we have analyzed the linear systems arising from the SSN method as well as from those appearing in the smooth modified Cahn–Hilliard equation. We have introduced and studied block-triangular preconditioners using an efficient and cheap to

apply Schur complement approximation. This approximation can be done using multilevel techniques, such as AMG (as in our case). In fact, we have found an optimal preconditioner for the smooth model that is independent of the mesh size. A comparison between the smooth and nonsmooth models has shown that the latter achieves better results.

The numerical results presented show that using a nonsmooth Cahn–Hilliard inpainting operator leads to a visually improved restored image when compared to existing inpainting tools. A major drawback of this approach was the need for a fast solution technique. In this paper we have designed and analyzed a reliable practical finite-element solver for this operator. Hence, we have presented a flexible, efficient, and effective inpainting tool.

**Acknowledgments.** The authors would like to thank Thomas Rolle for providing his photograph for this work and the anonymous referees for their helpful comments and suggestions.

## REFERENCES

- [1] W. BANGERTH, R. HARTMANN, AND G. KANSCHAT, *deal.II—a general purpose object oriented finite element library*, ACM Trans. Math. Software, 33 (2007), 24.
- [2] W. BANGERTH, T. HEISTER, AND G. KANSCHAT, *deal.II Differential Equations Analysis Library, Technical Reference*, <http://www.dealii.org>.
- [3] M. BERTALMÍO, A. L. BERTOZZI, AND G. SAPIRO, *Navier-Stokes, fluid dynamics, and image and video inpainting*, in Proceedings of the IEEE Computer Vision and Pattern Recognition (CVPR), IEEE Computer Society Press, Los Alamitos, CA, 2001, pp. 355–362.
- [4] M. BERTALMÍO, G. SAPIRO, V. CASELLES, AND C. BALLESTER, *Image inpainting*, in Proceedings of the ACM Computer Graphics and Interactive Techniques, SIGGRAPH, ACM Press/Addison–Wesley, New York, 2000, pp. 417–424.
- [5] A. L. 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.
- [6] 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.
- [7] J. F. BLOWEY AND C. M. ELLIOTT, *The Cahn–Hilliard gradient theory for phase separation with non-smooth free energy. I: Mathematical analysis*, European J. Appl. Math., 2 (1991), pp. 233–280.
- [8] J. F. BLOWEY AND C. M. ELLIOTT, *The Cahn–Hilliard gradient theory for phase separation with non-smooth free energy. II: Numerical analysis*, European J. Appl. Math., 3 (1992), pp. 147–179.
- [9] J. BOSCH, *Fast Solvers for Cahn–Hilliard Variational Inequalities*, Master’s thesis, University of Magdeburg, Magdeburg, Germany, 2012 (in German).
- [10] J. BOSCH, M. STOLL, AND P. BENNER, *Fast Solution of Cahn–Hilliard Variational Inequalities Using Implicit Time Discretization and Finite Elements*, Preprint MPIMD/13-01, Max Planck Institute Magdeburg, Magdeburg, Germany, 2013.
- [11] J. P. BOYD, *Chebyshev and Fourier Spectral Methods*, 2nd rev. ed., Dover, Mineola, NY, 2001.
- [12] 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.
- [13] 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.
- [14] T. CHAN AND J. SHEN, *Variational restoration of nonflat image features: Models and algorithms*, SIAM J. Appl. Math., 61 (2000), pp. 1338–1361.
- [15] 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.
- [16] T. F. CHAN AND J. SHEN, *Non-texture inpainting by curvature-driven diffusions (CDD)*, J. Vis. Commun. Image Represent., 12 (2001), pp. 436–449.
- [17] T. F. CHAN AND J. SHEN, *Mathematical models for local nontexture inpaintings*, SIAM J. Appl. Math., 62 (2002), pp. 1019–1043.

- [18] T. F. CHAN AND J. SHEN, *Variational image inpainting*, Comm. Pure Appl. Math., 58 (2005), pp. 579–619.
- [19] 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.
- [20] T. A. DAVIS, *UMFPACK Version 5.6.2 User Guide*, Technical Report TR-04-003 (revised), Department of Computer and Information Science and Engineering, University of Florida, Gainesville, FL, 2013.
- [21] I. C. DOLCETTA, S. F. VITA, AND R. MARCH, *Area-preserving curve-shortening flows: From phase separation to image processing*, Interfaces Free Bound., 4 (2002), pp. 325–343.
- [22] C. M. ELLIOTT, *The Cahn-Hilliard model for the kinetics of phase separation*, in Mathematical Models for Phase Change Problems, Internat. Ser. Numer. Math. 88, J. F. Rodrigues, ed., Birkhäuser, Basel, 1989, pp. 35–73.
- [23] C. M. ELLIOTT AND A. M. STUART, *The global dynamics of discrete semilinear parabolic equations*, SIAM J. Numer. Anal., 30 (1993), pp. 1622–1663.
- [24] C. M. ELLIOTT AND S. ZHENG, *On the Cahn-Hilliard equation*, Arch. Rational Mech. Anal., 96 (1986), pp. 339–357.
- [25] H. C. ELMAN, D. J. SILVESTER, AND A. J. WATHEN, *Finite Elements and Fast Iterative Solvers: With Applications in Incompressible Fluid Dynamics*, Oxford University Press, Oxford, UK, 2005.
- [26] S. ESEDOGLU AND J. H. SHEN, *Digital inpainting based on the Mumford-Shah-Euler image model*, European J. Appl. Math., 13 (2002), pp. 353–370.
- [27] D. J. EYRE, *An Unconditionally Stable One-Step Scheme for Gradient Systems*, Technical report, Department of Mathematics, University of Utah, Salt Lake City, UT, 1997.
- [28] D. J. EYRE, *Unconditionally gradient stable time marching the Cahn-Hilliard equation*, in Computational and Mathematical Models of Microstructural Evolution, Mater. Res. Soc. Symp. Proc. 529, J. W. Bullard, L.-Q. Chen, R. K. Kalia, and A. M. Stoneham, eds., Materials Research Society, Warrendale, PA, 1998, pp. 39–46.
- [29] R. D. FALGOUT, *An introduction to algebraic multigrid*, Comput. Sci. Eng., 8 (2006), pp. 24–33.
- [30] R. FLETCHER, *Conjugate gradient methods for indefinite systems*, in Numerical Analysis, Lecture Notes in Math. 506, G. A. Watson, ed., Springer, Berlin, Heidelberg, 1976, pp. 73–89.
- [31] R. W. FREUND AND N. M. NACHTIGAL, *QMR: A quasi-minimal residual method for non-Hermitian linear systems*, Numer. Math., 60 (1991), pp. 315–339.
- [32] H. GARCKE, *Mechanical effects in the Cahn-Hilliard model: A review on mathematical results*, in Mathematical Methods and Models in Phase Transitions, A. Miranville, ed., Nova Science, New York, 2005, pp. 43–77.
- [33] A. GILLETTE, *Image Inpainting Using a Modified Cahn-Hilliard Equation*, Ph.D. thesis, University of California, Los Angeles, CA, 2006.
- [34] A. GREENBAUM, *Iterative Methods for Solving Linear Systems*, Frontiers Appl. Math. 17, SIAM, Philadelphia, 1997.
- [35] W. HACKBUSCH, *Multigrid Methods and Applications*, Springer Ser. Comput. Math. 4, Springer, Berlin, 1985.
- [36] M. HEROUX, R. BARTLETT, V. H. R. HOEKSTRA, J. HU, T. KOLDA, R. LEHOUCQ, K. LONG, R. PAWLOWSKI, E. PHIPPS, A. SALINGER, H. THORNQUIST, R. TUMINARO, J. WILLENBRING, AND A. WILLIAMS, *An Overview of Trilinos*, Technical Report SAND2003-2927, Sandia National Laboratories, Albuquerque, NM, 2003.
- [37] 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.
- [38] M. HINTERMÜLLER, M. HINZE, AND M. H. TBER, *An adaptive finite-element Moreau-Yosida-based solver for a non-smooth Cahn-Hilliard problem*, Optim. Methods Softw., 26 (2011), pp. 777–811.
- [39] M. HINTERMÜLLER, K. ITO, AND K. KUNISCH, *The primal-dual active set strategy as a semismooth Newton method*, SIAM J. Optim., 13 (2003), pp. 865–888.
- [40] M. HINTERMÜLLER AND M. ULBRICH, *A mesh-independence result for semismooth Newton methods*, Math. Program., 101 (2004), pp. 151–184.
- [41] S. MASNOU AND J.-M. MOREL, *Level lines based disocclusion*, in Proceedings of the IEEE International Conference on Image Processing (ICIP), IEEE Computer Society Press, Chicago, IL, 1998, pp. 259–263.

- [42] D. MUMFORD, *Elastica and computer vision*, in Algebraic Geometry and Its Applications, C. L. Bajaj, ed., Springer, New York, 1994, pp. 491–506.
- [43] D. MUMFORD AND J. SHAH, *Optimal approximations by piecewise smooth functions and associated variational problems*, Comm. Pure Appl. Math., 42 (1989), pp. 577–685.
- [44] M. F. MURPHY, G. H. GOLUB, AND A. J. WATHEN, *A note on preconditioning for indefinite linear systems*, SIAM J. Sci. Comput., 21 (2000), pp. 1969–1972.
- [45] A. NOVICK-COHEN, *The Cahn-Hilliard equation: Mathematical and modeling perspectives*, Adv. Math. Sci. Appl., 8 (1998), pp. 965–985.
- [46] J. W. PEARSON AND M. STOLL, *Fast iterative solution of reaction-diffusion control problems arising from chemical processes*, SIAM J. Sci. Comput., 35 (2013), pp. B987–B1009.
- [47] J. W. PEARSON, M. STOLL, AND A. J. WATHEN, *Regularization-robust preconditioners for time-dependent PDE-constrained optimization problems*, SIAM J. Matrix Anal. Appl., 33 (2012), pp. 1126–1152.
- [48] J. W. PEARSON AND A. J. WATHEN, *A new approximation of the Schur complement in preconditioners for PDE-constrained optimization*, Numer. Linear Algebra Appl., 19 (2012), pp. 816–829.
- [49] J. W. PEARSON AND A. J. WATHEN, *Fast iterative solvers for convection-diffusion control problems*, Electron. Trans. Numer. Anal., 40 (2013), pp. 294–310.
- [50] L. I. RUDIN AND S. OSHER, *Total variation based image restoration with free local constraints*, in Proceedings of the IEEE International Conference on Image Processing (ICIP), IEEE Computer Society Press, Los Alamitos, CA, 1994, pp. 31–35.
- [51] L. I. RUDIN, S. OSHER, AND E. FATEMI, *Nonlinear total variation based noise removal algorithms*, Phys. D, 60 (1992), pp. 259–268.
- [52] J. W. RUGE AND K. STÜBEN, *Algebraic multigrid*, in Multigrid Methods, Frontiers Appl. Math. 3, SIAM, Philadelphia, 1987, pp. 73–130.
- [53] Y. SAAD, *Iterative Methods for Sparse Linear Systems*, 2nd ed., SIAM, Philadelphia, 2003.
- [54] Y. SAAD AND M. H. SCHULTZ, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 856–869.
- [55] C.-B. SCHÖNLIEB AND A. L. BERTOZZI, *Unconditionally stable schemes for higher order inpainting*, Commun. Math. Sci., 9 (2011), pp. 413–457.
- [56] G. STRANG AND G. FIX, *An Analysis of the Finite Element Method*, 2nd ed., Wellesley-Cambridge, Wellesley, MA, 2008.
- [57] A. TSAI, A. J. YEZZI, 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.
- [58] M. ULBRICH, *Semismooth Newton methods for operator equations in function spaces*, SIAM J. Optim., 13 (2003), pp. 805–841.
- [59] H. A. VAN DER VORST, *Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput., 13 (1992), pp. 631–644.
- [60] B. P. VOLLMAYR-LEE AND A. D. RUTENBERG, *Fast and accurate coarsening simulation with an unconditionally stable time step*, Phys. Rev. E (3), 68 (2003), 066703.
- [61] P. WESSELING, *An Introduction to Multigrid Methods*, John Wiley & Sons Ltd., Chichester, UK, 1992.
- [62] X.-F. WU AND Y. A. DZENIS, *Phase-field modeling of the formation of lamellar nanostructures in diblock copolymer thin films under inplanar electric fields*, Phys. Rev. E (3), 77 (2008), 031807.