A high-contrast fourth-order PDE from imaging: numerical solution by ADI splitting

Bertram Düring and Carola-Bibiane Schönlieb

ABSTRACT. We consider a nonlinear fourth-order diffusion equation that arises in denoising of image densities. We propose an alternative direction implicit (ADI) splitting method for its numerical solution. To treat the high-order and mixed derivative terms in the equation we adopt an ADI method by Hundsdorfer and Verwer to the present setting. The paper is furnished with numerical results for the evolution of simple densities and for image denoising.

1. Introduction

In this paper we consider the following nonlinear fourth-order evolution equation

$$(1.1) \qquad u_t = -\mathrm{div}\left(u\nabla\mathrm{div}\left(\frac{\nabla u}{|\nabla u|}\right)\right) \quad \text{in } \Omega \subset \mathbb{R}^2,$$

with an appropriate initial condition u_0 and periodic boundary conditions. We solve this equation numerically by an ADI splitting technique and discuss its scale space properties by means of numerical simulations.

This equation can be formally derived as the L^2 Wasserstein gradient flow

$$(1.2) \qquad \begin{aligned} u_t &= \mathrm{div}(u\nabla\mathcal{E}'(u)), \\ u(0,x) &= u_0(x) \geq 0, \end{aligned}$$

with normalized mass $\int_\Omega u_0\,dx = 1$ and the total variation functional $\mathcal{E}$ defined by [2, 11]

$$(1.3) \qquad \mathcal{E}(u) := |Du|(\Omega) = \sup_{\mathbf{g}\in C_0^\infty(\Omega;\mathbb{R}^d),\|g\|_\infty\leq 1}\int_\Omega u\,\nabla\cdot\mathbf{g}\,dx,$$

$d = 1, 2$. In [8] this equation first appeared in connection with density estimation and smoothing. Therein, the authors propose to compute a smoothed version u of a given probability density u_0 as a minimiser of

$$(1.4) \qquad \frac{1}{2}W_2(u_0\mathcal{L}^d, u\mathcal{L}^d)^2 + \alpha\mathcal{E}(u).$$

Here, $W_2(u_0\mathcal{L}^d, u\mathcal{L}^d)$ is the 2–Wasserstein distance between $u_0\mathcal{L}^d$ and $u\mathcal{L}^d$ ($\mathcal{L}^d$ denotes the Lebesgue measure in $\mathbb{R}^d$, $d = 1, 2$) and defines a distance within the space

---

of probability measures [**3, 20, 21, 1, 17**]. This minimisation problem can be interpreted as a discrete approximation of a solution of the gradient flow (1.1) of $\mathcal{E}(u)$ with respect to the L^2 Wasserstein metric. More precisely, the minimisation of (1.4) represents one timestep of De Giorgi's minimising movement scheme [**3, 16**] to the functional $\mathcal{E}(u)$ with timestep α. By construction the regularisation method (1.4) proposed in [**8**] is non-smooth, i.e., edge preserving, and conserves mass, i.e., is density preserving. In [**8**] the numerical solution of (1.4) has been done by a combination of the Benamou-Brenier formulation [**4**] for the Wasserstein distance, an augmented Lagrangian method and an operator splitting technique [**12, 6**]. This numerical procedure is in the flavour of several recently proposed numerical schemes for equations with gradient flow structure, cf., e.g., [**9, 5, 10, 7, 13**] and references therein.

In this paper we shall investigate equation (1.1) numerically by implementing an operator splitting technique. More specifically, we rewrite the fourth-order equation (1.1) as a system of two second-order equations. Then we discretise semi-implictly in time using a non-classical ADI-splitting in the spirit of [**15, 14**]. The latter is able to deal with differential equations involving mixed derivative terms. Note, that the proposed numerical scheme is not in the flavour of the gradient-flow solutions mentioned above. In fact, it does not use the Wasserstein-gradient framework to accomplish a solution of (1.1). Hence, it is not necessarily density preserving. We therefore monitor mass and positivity of our discrete solutions in our numerical experiments presented in section 3. We find that the solutions remain positive throughout and the initial mass is preserved (up to a relative numerical roundoff error of 10^{-6}).

The paper is organised as follows. In section 2 we revisit self-similar solutions of equation (1.1) in the continuum case. The main part is devoted to the presentation of the new operator splitting scheme to solve (1.1) numerically. After introducing the finite difference operators for the space discretisation in section 3.1, we present the fully discrete scheme in section 3.2. In section 3.3 the paper concludes with a numerical discussion focusing on the high-contrasting effects and multiscale properties of (1.1).

2. Self-similar solutions of the equation

To get a better understanding of the kind of solutions we can expect to receive from the evolution of (1.1) we first consider a special type of self-similar solutions that can be explicitly computed in one space dimension. In particular, in [**8**] the authors compute an interesting special class of solutions of self-similar structure of (1.4). The basis of the computations therein is the characterization of the optimal transport between a measure and its rescaled version, derived in [**4**], i.e., they look for solutions u that fulfill

$$(2.1) \qquad u_0(x)\ dx = \frac{1}{\delta^d}\ u\left(\frac{x}{\delta}\right)\ dx,$$

(a) Analytical self-similar solutions (2.2)

(b) Self-similar solutions of the TV-Wasserstein gradient flow numerically computed as in [**8**] for a Gaussian function as initial condition.
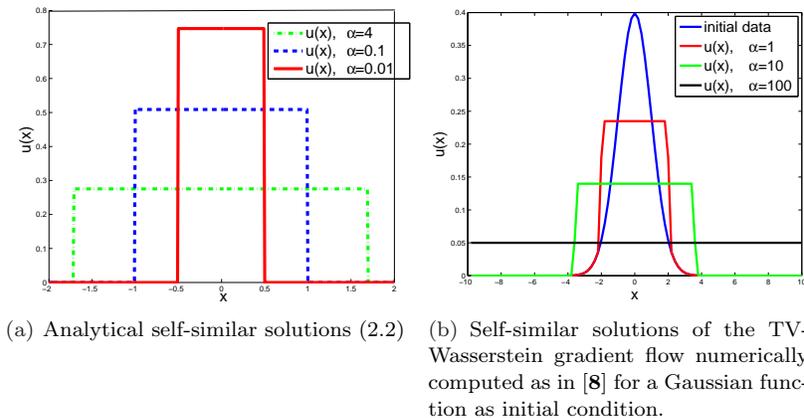
FIGURE 1. Self-similar Solutions for the total variation gradient flow.

with scaling parameter $\delta > 0$. In [**8**] the authors further assume $u(x) \geq 0$ in a neighbourhood around zero and consider the following ansatz for u

$$(2.2) \qquad u(x) = \begin{cases} \beta, & x \in (-x_0, x_0) \\ 0, & \text{otherwise,} \end{cases}$$

where $\beta > 0$ is a positive constant. This yields self-similar solutions of (1.4) of the form (2.2) with $\beta = 1/(2x_0)$ and $x_0 = \sqrt[6]{9\frac{\alpha}{a}}$, for a constant $a = a(\delta)$ and a regularisation parameter $\alpha > 0$. Please compare Figure 1(e) for a sketch of such solutions for different choices of α.

Equation (1.1) possesses the same type of self-similar solutions, where the evolution time takes on the role of the regularisation parameter α, cf. section 3.3 for numerical examples.

3. Numerical solution with ADI splitting

3.1. Discretisation. For the numerical implementation we need to approximate the derivatives in the differential equations involved in our models. We consider finite differences for this and use periodic boundary conditions. Let $\Omega = [0,1]^2$ be the two-dimensional space domain that we consider for our problem. Let $T > 0$ denote the final time for the evolution. We partition the space-time-cylinder $[0,1]^2 \times [0,T]$ into equidistant cells $[x_i, x_{i+1}] \times [y_j, y_{j+1}]$, $i, j = 0, \ldots, N-1$, with $x_i = ih$ and $y_j = jh$ and step-size $h = 1/N$, and time intervals $[t_n, t_{n+1}]$ for $n = 0, \ldots, M-1$ with $t_n = \Delta t\, n$ and time-steps $\Delta t = \frac{T}{M}$. Let u be an arbitrary (sufficiently smooth) function defined on $\Omega \times [0,T]$, then we denote with $U_{i,j}^n \approx u(x_i, y_j, t_n)$ the approximation of u in the node (x_i, y_j) at time level t_n. We discretise all first derivatives with central differences with periodic boundary conditions. More precisely, we approximate $(D_x^o U^n)_{i,j} \approx u_x(x_i, y_j, t_n)$, where

$$(3.1) \qquad (D_x^o U)_{i,j} = \begin{cases} \frac{u_{i+1,j} - u_{i-1,j}}{2h} & i = 1, \ldots, N-1 \\ \frac{u_{1,j} - u_{N-1,j}}{2h} & i = 0, N. \end{cases}$$

The first derivative of u with respect to y is approximated analogously by $(D_y^o U)_{i,j}$ and we write $D_\nabla^o U$ for the corresponding approximation of the gradient vector ∇u. The second derivatives are either approximated by the five-point formula, e.g., the Laplace operator $\Delta u = u_{xx} + u_{yy}$ is approximated by

(3.2)
$$(D_\Delta U)_{i,j} = (D_{xx} U)_{i,j} + (D_{yy} U)_{i,j} = \frac{u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1} - 4u_{i,j}}{h^2},$$

or by a smoother finite difference approximation that averages over every second grid point only, i.e.,

(3.3)
$$(D_\Delta^o U)_{i,j} = (D_{xx}^o U)_{i,j} + (D_{yy}^o U)_{i,j} = \frac{u_{i+2,j} + u_{i-2,j} + u_{i,j+2} + u_{i,j-2} - 4u_{i,j}}{4h^2}.$$

Both discrete Laplacians are second-order accurate.

3.2. The ADI scheme. In the following we shall apply an ADI splitting scheme to the numerical solution of (1.1). To do so we first regularise the equation by replacing $|\nabla u|$ by $|\nabla u|_\epsilon := \sqrt{u_x^2 + u_y^2 + \epsilon}$, with $0 < \epsilon \ll 1$, and get

(3.4)
$$u_t = -\mathrm{div}\left(u\nabla\mathrm{div}\left(\frac{\nabla u}{|\nabla u|_\epsilon}\right)\right).$$

This is a common procedure for evolution equations of the total variation functional, e.g., cf. [**19, 18**]. One possibility to apply an ADI-splitting to (3.4) would be to adopt the method in [**22**], where the authors consider ADI splitting for fourth-order nonlinear equations like the thin-film equation together with an outer Newton-iteration.

Here, we rewrite the fourth-order evolution equation (3.4) as a system of two second-order equations for (u,v)

$$u_t = \mathrm{div}\left(u\nabla v\right),$$

$$v = -\mathrm{div}\left(\frac{\nabla u}{|\nabla u|_\epsilon}\right).$$

We approximate the nonlinear differential operators by linear ones in the following way

(3.5)
$$u_t = \nabla u \cdot \nabla \tilde{v} + \tilde{u}\Delta v,$$

(3.6)
$$v = -\frac{\epsilon + \tilde{u}_y^2}{|\nabla\tilde{u}|_\epsilon^3}u_{xx} - \frac{\epsilon + \tilde{u}_x^2}{|\nabla\tilde{u}|_\epsilon^3}u_{yy} + 2\frac{\tilde{u}_x\tilde{u}_y}{|\nabla\tilde{u}|_\epsilon^3}u_{xy},$$

where $(\tilde{u}, \tilde{v})$ is a given pair of functions, which will be the solution in the old time step in the ADI scheme below. Using the notation from section 3.1 we replace the continuous space derivatives with its finite difference approximations as follows

(3.7)
$$U_t = D_\nabla^o U \cdot D_\nabla^o \tilde{V} + \tilde{U} D_\Delta^o V,$$

(3.8)
$$V = -\frac{\epsilon + (D_y^o \tilde{U})^2}{|D_\nabla^o \tilde{U}|_\epsilon^3}D_{xx}U - \frac{\epsilon + (D_x^o \tilde{U})^2}{|D_\nabla^o \tilde{U}|_\epsilon^3}D_{yy}U + 2\frac{D_x^o \tilde{U} D_y^o \tilde{U}}{|D_\nabla^o \tilde{U}|_\epsilon^3}D_{xy}U,$$

where (U,V) is the semi-discrete approximation to a solution of (3.5)-(3.6). Note that the Laplace operator applied to v in the first equation (3.5) is approximated by the second-order finite difference approximation (3.3). This is a smoother version of

the 5-point formula (3.2) and turned out to be necessary for a stable approximation of (1.1).

With the above finite difference matrices, the system of equations (3.7)-(3.8) can be written as

$$\begin{pmatrix} U_t \\ V \end{pmatrix} = F(U,V) = \begin{pmatrix} A & B \\ C & D \end{pmatrix} \cdot \begin{pmatrix} U \\ V \end{pmatrix},$$

with corresponding matrices $A, B, C, D \in \mathbb{R}^{N^2 \times N^2}$. Then, $F(U,V)$ is split into its mixed derivative terms F_0, and its terms F_1 and F_2 with derivatives with respect to x and y only, respectively. This gives

$$F(U,V) = F_0(U,V) + F_1(U,V) + F_2(U,V),$$

with

$$F_0(U,V) = \begin{pmatrix} A_0 & B_0 \\ C_0 & D_0 \end{pmatrix} \cdot \begin{pmatrix} U \\ V \end{pmatrix} = \begin{pmatrix} \mathcal{O} & \mathcal{O} \\ 2\frac{D_x^o \tilde{U} D_y^o \tilde{U}}{|D_\nabla^o \tilde{U}|_\epsilon^3} D_{xy} & \mathcal{O} \end{pmatrix} \cdot \begin{pmatrix} U \\ V \end{pmatrix},$$

$$F_1(U,V) = \begin{pmatrix} A_1 & B_1 \\ C_1 & D_1 \end{pmatrix} \cdot \begin{pmatrix} U \\ V \end{pmatrix} = \begin{pmatrix} D_x^o \tilde{V} D_x^o & \tilde{U} D_{xx}^o \\ -\frac{\epsilon + (D_y^o \tilde{U})^2}{|D_\nabla^o \tilde{U}|_\epsilon^3} D_{xx} & \mathcal{O} \end{pmatrix} \cdot \begin{pmatrix} U \\ V \end{pmatrix},$$

$$F_2(U,V) = \begin{pmatrix} A_2 & B_2 \\ C_2 & D_2 \end{pmatrix} \cdot \begin{pmatrix} U \\ V \end{pmatrix} = \begin{pmatrix} D_y^o \tilde{V} D_y^o & \tilde{U} D_{yy}^o \\ -\frac{\epsilon + (D_x^o \tilde{U})^2}{|D_\nabla^o \tilde{U}|_\epsilon^3} D_{yy} & \mathcal{O} \end{pmatrix} \cdot \begin{pmatrix} U \\ V \end{pmatrix}.$$

For an initial condition (U^0, V^0) we now look for approximate solutions $(U^n, V^n) \approx (u(t_n), v(t_n))$ where $t_n = n\Delta t$, $n = 1, 2, \ldots$, of (3.5)-(3.6). With $(\tilde{U}, \tilde{V}) = (U^{n-1}, V^{n-1})$ and adapting the ADI scheme from [14] to our setting we compute these approximate solutions as

$$(1) \quad \begin{pmatrix} Y_0^1 \\ Y_0^2 \end{pmatrix} = \begin{pmatrix} U^{n-1} + \Delta t F^1(U^{n-1}, V^{n-1}) \\ F^2(Y_0^1, V^{n-1}) \end{pmatrix},$$

$$(2) \quad \begin{pmatrix} Y_1^1 \\ Y_1^2 \end{pmatrix} = \begin{pmatrix} Y_0^1 \\ Y_0^2 \end{pmatrix} + \begin{pmatrix} \frac{1}{2}\Delta t(F_1^1(Y_1^1, Y_1^2) - F_1^1(U^{n-1}, V^{n-1})) \\ F_1^2(Y_1^1, Y_1^2) - F_1^2(U^{n-1}, V^{n-1}) \end{pmatrix},$$

$$(3) \quad \begin{pmatrix} Y_2^1 \\ Y_2^2 \end{pmatrix} = \begin{pmatrix} Y_1^1 \\ Y_1^2 \end{pmatrix} + \begin{pmatrix} \frac{1}{2}\Delta t(F_2^1(Y_2^1, Y_2^2) - F_2^1(U^{n-1}, V^{n-1})) \\ F_2^2(Y_2^1, Y_2^2) - F_2^2(U^{n-1}, V^{n-1}) \end{pmatrix},$$

$$(4) \quad \begin{pmatrix} \tilde{Y}_0^1 \\ \tilde{Y}_0^2 \end{pmatrix} = \begin{pmatrix} Y_0^1 + \frac{1}{2}\Delta t(F^1(Y_2^1, Y_2^2) - F^1(U^{n-1}, V^{n-1})) \\ F^2(\tilde{Y}_0^1, V^{n-1}) \end{pmatrix},$$

$$(5) \quad \begin{pmatrix} \tilde{Y}_1^1 \\ \tilde{Y}_1^2 \end{pmatrix} = \begin{pmatrix} \tilde{Y}_0^1 \\ \tilde{Y}_0^2 \end{pmatrix} + \begin{pmatrix} \frac{1}{2}\Delta t(F_1^1(\tilde{Y}_1^1, \tilde{Y}_1^2) - F_1^1(Y_2^1, Y_2^2)) \\ F_1^2(\tilde{Y}_1^1, \tilde{Y}_1^2) - F_1^2(Y_2^1, Y_2^2) \end{pmatrix},$$

$$(6) \quad \begin{pmatrix} U^n \\ V^n \end{pmatrix} = \begin{pmatrix} \tilde{Y}_1^1 \\ \tilde{Y}_1^2 \end{pmatrix} + \begin{pmatrix} \frac{1}{2}\Delta t(F_2^1(\tilde{Y}_2^1, \tilde{Y}_2^2) - F_2^1(Y_2^1, Y_2^2)) \\ F_2^2(\tilde{Y}_2^1, \tilde{Y}_2^2) - F_2^2(Y_2^1, Y_2^2) \end{pmatrix}.$$

Here, the superindices of the F terms, i.e., F^j, $j = 1, 2$, denote the first or second row of the respective coefficient matrix. The first (1) and fourth (4) computation of the algorithm are explicit time-steps taking care of the mixed derivative terms. The second (2) and fifth (5), and the third (3) and sixth (6) computations are implicit time-steps for the terms with pure x- and pure y-derivatives, respectively. These implicit steps consist only of tridiagonal systems and are therefore inexpensive to solve. Additionally, the matrices from step (2) and (3) reappear in (5) and (6) with different right hand sides, respectively. Hence, they have to be factorized only once. Overall, in every time step of the scheme, only two tridiagonal systems are solved.

3.3. Numerical results. We conclude the paper with a presentation of numerical results achieved with the ADI discretisation presented above. Special focus is laid on the presentation of the high-contrasting and multiscale properties of the equation. Due to the nonlinearity introduced by the total variation the equation preserves and enhances sharp features in the solution. This results in a high-contrasting effect. Moreover, multiple scales in the initial condition are apparent at different times of the evolution. This is called scale space of the differential equation.
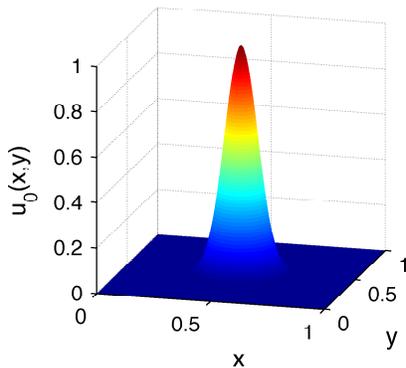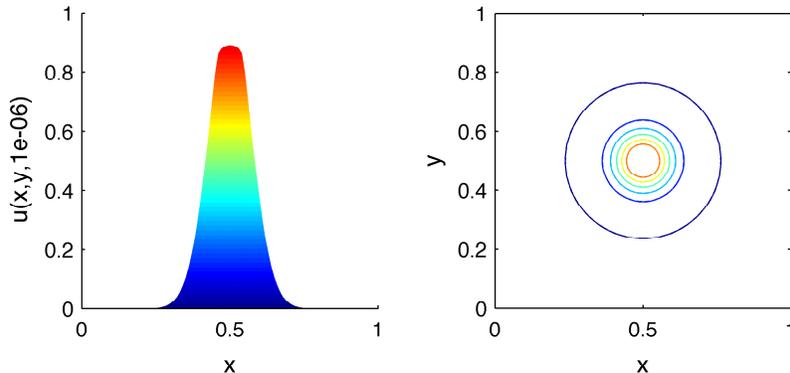

(a) Initial condition U^0

(b) Solution U^n at time $t_n = 10^{-6}$ with $\epsilon = 1$ (c) Level lines of U^n at time $t_n = 10^{-6}$ with $\epsilon = 1$

FIGURE 2. Initial condition and solution of the nonlinear fourth order total variation flow (1.1) for $\epsilon = 1$ at time 10^{-6}.

In all examples the computational domain Ω is the unit square with 100×100 gridpoints. In Figures 2-4 the effect of the evolution (1.1) for a simple two-dimensional density function is analysed. The initial condition U^0 in Figure 2(a) equals a Gaussian density $U_{i,j}^0 = \exp\left(((x_i - 1/2)^2 + (y_j - 1/2)^2)/\sigma^2\right)$ with variance $\sigma^2 = 0.01$. As expected from the structure of the self-similar solutions (2.2) discussed in section 2, as time progresses the top of the Gaussian is flattened, the amplitude decreases and its support becomes larger. The qualitative behaviour of

the solution in our examples is very similar to the self-similar solution of (1.4) for $\alpha = 1$, cf. Figure 1(b). The solutions are computed for different choices of regularising parameters ϵ in (3.4). It can be seen that the sharpness of the ridge between flat and steep part of the density increases with smaller ϵ, cf. e.g., the solution U^n at time $t_n = 10^{-6}$ in Figure 2(b-c) and in Figure 3(a-b).

We have also monitored mass and positivity of the discrete solutions in all our simulations. We find that the solutions remain positive throughout the evolution. The initial mass is preserved over time up to a relative numerical roundoff error of 10^{-6} which is well below the discretisation error.

Motivated by its original application in smoothing of density images [8], in Figure 5 the denoising and scale space properties of equation (1.1) are numerically simulated for the given noisy image in Figure 5(a). Due to the nonlinear nature of the equation the diffusion is anisotropic. More precisely, the diffusion coefficient depends on the reciprocal of the size of the image gradient and as such diffuses more in homogeneous areas of the image and less at image edges, cf. Figures 5(b-c). To measure the quality of the denoised image we consider the signal-to-noise ratio (SNR) computed as SNR $= 20 \log \left[\sum_{i,j} U_{i,j}^2 / \sum_{i,j} (U_{org} - U)_{i,j}^2 \right]^{1/2}$, where U_{org} is the original image without noise. Starting with the noisy image in Figure 5(a) with SNR $= 36.15$ this value improves for the solution at $t_n = 10^{-7}$ to SNR$= 37.18$, cf. Figure 5(b). For the next solution plotted at time $t_n = 10^{-6}$ in Figure 5(c) the solution has already lost on its quality improvement with SNR$=33.56$. This is due to the multiscale properties of the equation, i.e. as time evolves the small scales in the image get more and more absorbed by the larger scales, cf. Figure 5(c-e). These larger scales are the edges that outline the main structures in the image, e.g, the black boundary of the box and the contour of the writing '2012'. As time evolves further the structures are smoothed out even more (see Figure 5(f)) and eventually the solution will converge to a constant steady state (average of all grayvalues in U^0).

Acknowledgements

References

[1] Ambrosio, L.; Caffarelli, L. A.; Brenier, Y: *Optimal Transportation and Applications* Lecture Notes in Mathematics, Springer-Verlag Berlin Heidelberg, (2003).

[2] Ambrosio, L.; Fusco, N.; Pallara, D.: *Functions of Bounded Variation and Free Discontinuity problems*, Oxford Mathematical Monographs. Oxford: Clarendon Press. xviii, (2000).

[3] Ambrosio, L.; Gigli, N.; Savare, G.: *Gradient Flows in Metric Spaces and in the Space of Probability Measures*, Lectures in Mathematics ETH Zrich, Birkhäuser Verlag, Basel, (2005).

[4] Benamou, J. D.; Brenier, Y.: *A computational fluid mechanics solution to the Monge- Kantorovich mass transfer problem*, Numer. Math. 84 , 375-393, (2000).

[5] Blanchet, A.; Calvez, V.; Carrillo, J. A.: *Convergence of the mass-transport steepest descent scheme for the sub-critical Patlak-Keller-Segel model*, SIAM J. Numer. Anal. 46, pp. 691–721, (2008).
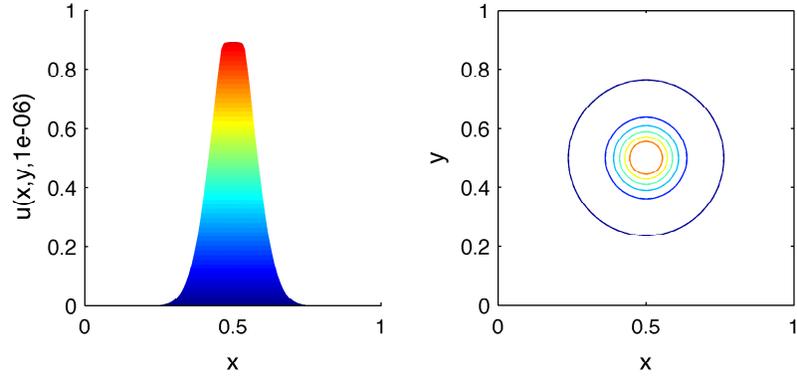
(a) Solution U^n at time $t_n = 10^{-6}$ with $\epsilon = 0.1$ (b) Level lines of U^n at time $t_n = 10^{-6}$ with $\epsilon = 0.1$
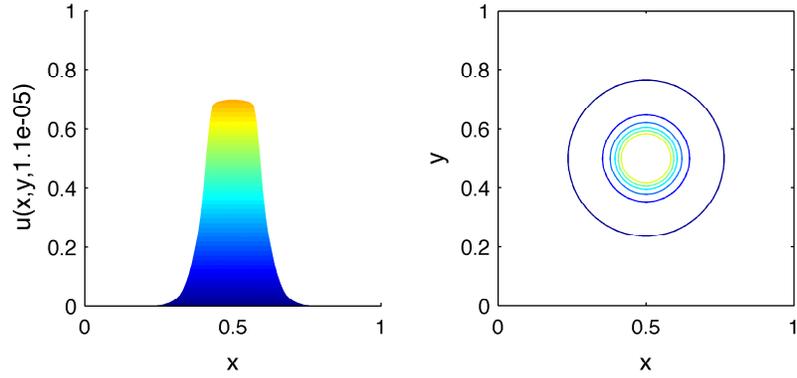
(c) Solution U^n at time $t_n = 1.1 \cdot 10^{-5}$ with $\epsilon = 0.1$ (d) Level lines of U^n at time $t_n = 1.1 \cdot 10^{-5}$ with $\epsilon = 0.1$
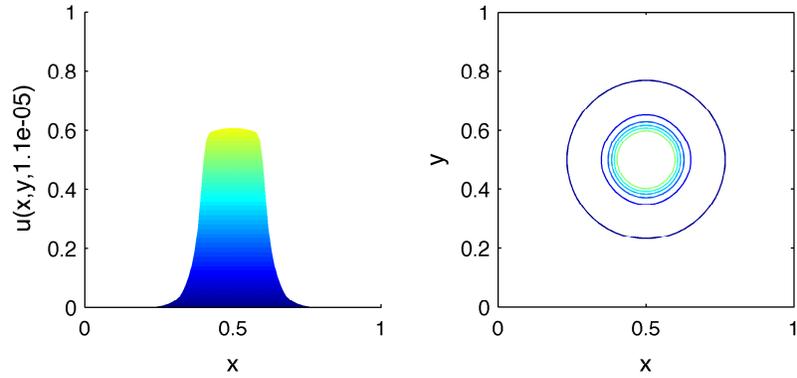
(e) Solution U^n at time $t_n = 3.1 \cdot 10^{-5}$ with $\epsilon = 0.1$ (f) Level lines of U^n at time $t_n = 3.1 \cdot 10^{-5}$ with $\epsilon = 0.1$

FIGURE 3. Solution of (1.1) for $\epsilon = 0.1$ with initial condition U^0 (cf. Figure 2(a)) at time 10^{-6}.

(a) Solution U^n at time $t_n = 10^{-6}$ with $\epsilon = 0.01$

(b) Level lines of U^n at time $t_n = 10^{-6}$ with $\epsilon = 0.01$

(c) Solution U^n at time $t_n = 10^{-5}$ with $\epsilon = 0.01$

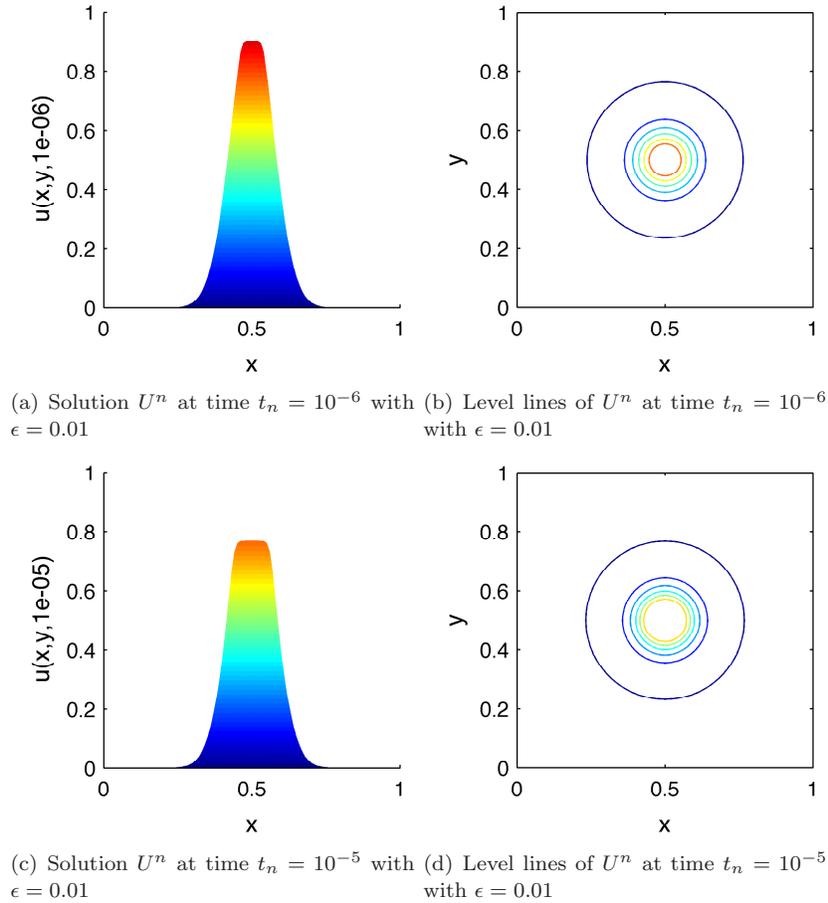(d) Level lines of U^n at time $t_n = 10^{-5}$ with $\epsilon = 0.01$

FIGURE 4. Solution of (1.1) for $\epsilon = 0.01$ with initial condition U^0 (cf. Figure 2(a)) at times 10^{-6} and 10^{-5}.

[6] Brune, C.: *4D Imaging in Tomography and Optical Nanoscopy*, PhD Thesis, University of Münster, July (2010).

[7] Burger, M.; Carrillo, J. A.; Wolfram, M.-T.: *A Mixed Finite Element Method for Nonlinear Diffusion Equations*, Kinetic and Related Models 3, 59–83, (2010).

[8] Burger, M.; Franek, M.; Schönlieb, C.-B.: *Regularised Regression and Density estimation based on Optimal Transport*, AMRX, to appear, (2012).

[9] Carrillo, J. A.; Moll, J. S.: *Numerical simulation of diffusive and aggregation phenomena in nonlinear continuity equations by evolving diffeomorphisms*, SIAM J. Sci. Comput. 31, pp. 4305-4329, (2009).

[10] Düring, B.; Matthes, D.; Milišić, J.-P.: *A gradient flow scheme for nonlinear fourth order equations*, Discrete Contin. Dyn. Syst. Ser. B 14(3), pp.935-959, (2010).

[11] Evans, L. C.; Gariepy, R. F.: *Measure Theory and Fine Properties of Functions.*, CRC Press, (1992).

[12] Goldstein, T.; Osher, S.: *The Split Bregman Method for L1 Regularized Problems.* SIAM Journal on Imaging Sciences 2(2), pp. 323–343, (2009).

[13] Gosse, L.; Toscani, G.: *Lagrangian numerical approximations to one-dimensional convolution-diffusion equations*, SIAM J. Sci. Comput., 28 (4) pp. 1203-1227, (2006).

(a) Initial condition U^0 (b) Solution U^n at time $t_n = 10^{-7}$ with $\epsilon = 0.1$ (c) Solution U^n at time $t_n = 10^{-6}$ with $\epsilon = 0.1$

(d) Solution U^n at time $t_n = 2 \cdot 10^{-6}$ with $\epsilon = 0.1$ (e) Solution U^n at time $t_n = 3 \cdot 10^{-6}$ with $\epsilon = 0.1$ (f) Solution U^n at time $t_n = 10^{-5}$ with $\epsilon = 0.1$
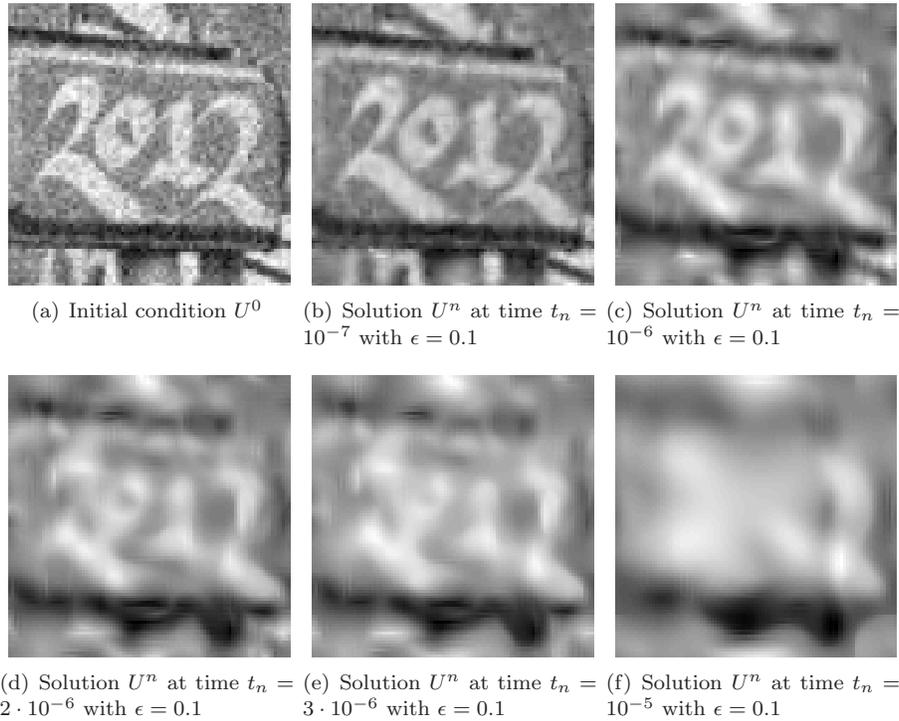
FIGURE 5. Scale space for image denoising. The initial condition U^0 in (a) is an image with additive Gaussian noise of variance 0.1. In (b)-(c) denoised images are computed as solutions of the nonlinear fourth order total variation flow (1.1) for $\epsilon = 0.1$ and at times 10^{-7}, 10^{-6}, $2 \cdot 10^{-6}$, 3×10^{-6} and 10^{-5}.

[14] In 't Hout, K.J.; Welfert, B.D.: Stability of ADI schemes applied to convection-diffusion equations with mixed derivate terms, *Applied Numerical Mathematics*, **57**, 19–35, (2007).

[15] Hundsdorfer, W.; Verwer, J.G.: Numerical solution of time-dependent advection-diffusion-reaction equations, Springer Series in Computational Mathematics, **33**, Springer-Verlag, Berlin, (2003).

[16] Jordan, R.; Kinderlehrer, D.; Otto, F.: *The variational formulation of the Fokker-Planck equation*, SIAM J. Math. Anal 29, pp. 1–17, (1999).

[17] Kantorovich, L. V.: *On a problem of Monge.* Uspekhi Mat. Nauk. 3, 225-226, (1948).

[18] Osher, S.; Burger, M.; Goldfarb, D.; Xu, J.; Yin, W.: *An iterative regularisation method for total variation-based image restoration.* Simul. 4, pp. 460–489, (2005).

[19] Rudin, L. I.; Osher, S.J.; Fatemi, E.: *Nonlinear total variation based noise removal algorithms*, Physica D **60** (1992), 259-268.

[20] Villani, C.: *Topics in optimal transportation*, vol. 58 of Graduate Studies in Mathematics, American Mathematical Society, Providence, RI, (2003).

[21] Villani, C.: *Optimal transport. Old and new*, Grundlehren der Mathematischen Wissenschaften, 338. Springer-Verlag, Berlin, (2009).

[22] Witelski, T.P.; Bowen, M.: *ADI schemes for higher-order nonlinear diffusion equations*, Appl. Numer. Math., 45, 331-351, (2003).

Department of Mathematics, University of Sussex, Pevensey II, Brighton, BN1 9QH, United Kingdom

E-mail address: `b.during@sussex.ac.uk`

DAMTP, Centre for Mathematical Sciences, Wilberforce Road, Cambridge, CB3 0WA, United Kingdom

E-mail address: `C.B.Schoenlieb@damtp.cam.ac.uk`