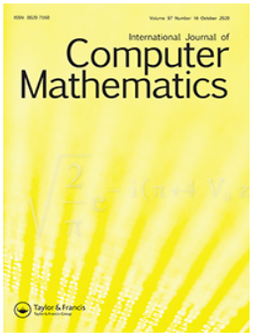


International Journal of Computer Mathematics

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/gcom20

On an effective multigrid solver for solving a class of variational problems with application to image segmentation

Michael Roberts , Ke Chen , Jingzhi Li & Klaus L. Irion

To cite this article: Michael Roberts , Ke Chen , Jingzhi Li & Klaus L. Irion (2020) On an effective multigrid solver for solving a class of variational problems with application to image segmentation, International Journal of Computer Mathematics, 97:10, 2015-2035, DOI: 10.1080/00207160.2019.1674290

To link to this article: https://doi.org/10.1080/00207160.2019.1674290

Accepted author version posted online: 30 Sep 2019.
Published online: 10 Oct 2019.

INTERNATIONAL JOURNAL OF COMPUTER MATHEMATICS
2020, VOL. 97, NO. 10, 2015–2035
https://doi.org/10.1080/00207160.2019.1674290

Taylor & Francis
Taylor & Francis Group

ARTICLE

On an effective multigrid solver for solving a class of variational problems with application to image segmentation

Michael Roberts[a], Ke Chen[a], Jingzhi Li[b] and Klaus L. Irion[c]

[a]Centre for Mathematical Imaging Techniques and Department of Mathematical Sciences, University of Liverpool, Liverpool, UK; [b]Department of Mathematics, Southern University of Science and Technology, Shenzhen, People's Republic of China; [c]Department of Radiology, Manchester University NHS Foundation Trust, Manchester, UK

ABSTRACT

In this paper we reformulate a class of non-linear variational models for global and selective image segmentation and obtain convergent multigrid solutions. In contrast, non-linear multigrid schemes do not converge for these problems with strong non-linearity and non-smoothness (jumps). Our new approach is to reformulate the non-linear models, using splitting techniques, to generate linear models in a higher dimension which are easier to solve and amenable to the linear multigrid framework. Although splitting techniques are well studied in isolation, direct application of a splitting idea is not sufficient and it is the combination of two splitting approaches and linear multigrid theory approaches which results in a highly effective multigrid algorithm. Numerical results demonstrate the fast convergence of the new multigrid methods.

ARTICLE HISTORY
Received 1 October 2018
Revised 6 September 2019
Accepted 25 September 2019

KEYWORDS
Iterative methods; multigrid; Euler-Lagrange equations; segmentation models

AMS SUBJECT CLASSIFICATIONS
65F10; 65N55; 35J20; 49Q10

1. Introduction

Non-linear systems of algebraic equations which arise from the numerical solution of partial differential equations (PDEs) are ubiquitous in computational science and engineering. In imaging sciences, demand for sharp contrast in images leads to strong non-linearity which in turn provides a rich source of non-linear problems to study.

In this paper, we are primarily concerned with (non-linear and non-smooth) variational models for image segmentation problems. For the class of problems that we consider here, there are currently no convergent multigrid algorithms. Although the non-linear multigrid Full Approximation Scheme (FAS) of Brandt [12] has been successfully applied to many PDEs (see e.g. [8,54]), the non-linear coefficients must be smooth to achieve convergence of FAS with standard smoothing schemes.

In this paper, we develop a fast and convergent multigrid solver for the PDEs which result from minimization of a class of variational image segmentation models. The key to this development is based on the classical idea in inverse problems of using splitting, rather than solving a hard problem directly; we split it into several easier subproblems. In our case we reformulate the problem for our class of non-linear models from a non-linear multigrid framework into four problems within a linear multigrid framework. The benefit of this is that for the linear multigrid framework there is extensive research into convergence guarantees, examples being [10–12,14,16,34,46,47,63,66,68], and only sparse literature in the non-linear multigrid framework, an almost exhaustive list being [7,34–36,40,54,55,60,67]. The reformulation step is accomplished through combining two distinct

CONTACT Ke Chen k.chen@liverpool.ac.uk; [web www.liv.ac.uk/ ~ cmchenke]

splitting ideas, those of Bresson *et al.* [15] and Goldstein *et al.* [30]. These splittings are individually well studied, however it is only by combining them in the right way that we achieve far simpler problems which can be solved with a convergent algorithm. The reformulation ultimately leads to solving 4 problems rather than 1, however 3 of these have closed form solutions and the other is solved iteratively. The proposed formulation is found to be fast to converge.

The rest of the paper is organized as follows. In Section 2 we review the current variational image segmentation models and some fast solvers. In Section 3 we discuss the class of models we wish to study and the problems associated with implementing the non-linear multigrid for this class of models. In Section 4 our two proposed optimization models will be introduced and in Section 5 the associated single level and multi-level algorithms will be introduced. In Section 6 we give numerical results comparing all algorithms and we finish in Section 7 with some concluding remarks.

Throughout this paper we denote the image domain by $\Omega = [0,1]^2 \subset \mathbb{R}^2$ and the image we would like to segment is denoted by $z(\boldsymbol{x}) : \Omega \to \mathbb{R}$. Let z be a given image defined in $\Omega \subset \mathbb{R}^2$.

2. Variational image segmentation models

Variational techniques for image segmentation involve minimizing an energy functional to find the boundary, Γ, of an object defined in the image domain Ω. We employ the level set framework of [50] whereby Γ will be found as the zero level set of function ϕ i.e. $\Gamma = \{\boldsymbol{x} \,|\, \phi(\boldsymbol{x}) = 0\}$. Then the inside of the objects is given by $\{\phi(\boldsymbol{x}) > 0\}$ and the outside by $\{\phi(\boldsymbol{x}) < 0\}$. Essentially, the key quantity is the Heaviside function

$$H(\phi) = \begin{cases} 1, & \phi \geq 0, \\ 0, & \phi < 0, \end{cases} \tag{1}$$

where the inside of the objects is characterized by the Heaviside taking value 1. We note that $H(\phi)$ is non-convex in ϕ and, therefore, if a model contains $H(\phi)$ in its formulation it is non-convex and challenging to solve.

We can follow the technique of Chan *et al.* [23] who replace $H(\phi)$ by a binary function $u \in \{0,1\}$ and then relax the constraint to $u \in [0,1]$. Using this technique, one can reformulate minimization with respect to ϕ to a new convex problem minimizing u.

Further the global minimizer of the energy functional is given by $u = \chi_\Sigma$ where $\Sigma = \{\boldsymbol{x} \,|\, u(\boldsymbol{x}) > \gamma\}$ and χ is the characteristic function. It is proven in [23] that this is a global minimizer for almost every $\gamma \in (0,1)$. In this case we find the boundary of the objects as the γ level set $\Gamma = \{\boldsymbol{x} \,|\, u(\boldsymbol{x}) = \gamma\}$. In this paper we set $\gamma = 0.5$ throughout.

We now briefly review some relevant models before we focus on fast solution issues.

2.1. The Chan-Vese model

Chan and Vese [22], introduced one of the most widely used and studied global image segmentation models, given by

$$\begin{aligned} \min_{\phi,c_1,c_2} F_{CV0}(\phi,c_1,c_2) = \mu \int_\Omega g(|\nabla z(\boldsymbol{x})|)|\nabla H_\varepsilon(\phi)| \,\mathrm{d}\Omega \\ + \lambda_1 \int_\Omega (z(\boldsymbol{x}) - c_1)^2 H_\varepsilon(\phi) \,\mathrm{d}\Omega + \lambda_2 \int_\Omega (z(\boldsymbol{x}) - c_2)^2 (1 - H_\varepsilon(\phi)) \,\mathrm{d}\Omega, \end{aligned} \tag{2}$$

where $g(s) = 1/(1+\beta s^2)$ is an edge detector function [19], β is a fixed non-negative tuning parameter and $H_\varepsilon(\phi)$ is a regularized version of $H(\phi)$ for small $\varepsilon > 0$ [22]. The values c_1 and c_2 are the average pixel intensities inside and outside of Γ. The terms μ, λ_1 and λ_2 are fixed non-negative tuning parameters.

The non-convex model (2) can be reformulated, by the convex relaxation technique of Chan *et al.* [23], to the convex model

$$\min_{u,c_1,c_2} F_{CV1}(u,c_1,c_2) = \mu \int_\Omega g(|\nabla z(\boldsymbol{x})|)|\nabla u|\,\mathrm{d}\Omega + \int_\Omega \left[\lambda_1(z(\boldsymbol{x})-c_1)^2 - \lambda_2(z(\boldsymbol{x})-c_2)^2\right] u\,\mathrm{d}\Omega + \alpha \int_\Omega \nu(u)\,\mathrm{d}\Omega, \tag{3}$$

where the exact penalty function $\nu(u) = \max\{0,\ 2|u-1/2|-1\}$ [23,61] is used to ensure $u \in [0,1]$, with the requirement that $\alpha > \frac{1}{2}\left|\left|\lambda_1(z(\boldsymbol{x})-c_1)^2 - \lambda_2(z(\boldsymbol{x})-c_2)^2\right|\right|$. The Euler-Lagrange equation for this model is given by

$$\mu\nabla\cdot\left(g(|\nabla z(\boldsymbol{x})|)\frac{\nabla u}{|\nabla u|_{\varepsilon_1}}\right) - \left[\lambda_1(z(\boldsymbol{x})-c_1)^2 - \lambda_2(z(\boldsymbol{x})-c_2)^2\right] - \alpha\nu'_{\varepsilon_2}(u) = 0, \tag{4}$$

with Neumann boundary condition $\partial u/\partial \boldsymbol{n} = 0$ for $\boldsymbol{n}$ the unit outward normal. Notice that we have introduced two small regularity parameters ε_1 [23] and ε_2 [61] at this point to avoid instabilities in the numerical calculations when solving the PDE. The former is used in the quantity $|\nabla u|_{\varepsilon_1} = \sqrt{u_x^2 + u_y^2 + \varepsilon_1}$ to avoid a zero denominator where $\nabla u = 0$ and the latter is used to smooth the kinks in the formulation of $\nu(u)$ (see [61] for details). The accurate choice of the parameter ε_1 is critical for convergence of the non-linear multigrid iterative solver [2,17,21,28], and is one of the main motivations for this paper. This problem is discussed at the end of Section 2.

2.2. Some selective segmentation models

In contrast to the global segmentation models which segment all objects in an image, selective segmentation models isolate an individual object, or objects. This is accomplished by adding a distance term $\mathcal{D}$ in the energy functional which penalizes pixels which are far from a user defined marker set $\mathcal{M}$. This ensures the final segmentation result is near to the user input.

There is a vast zoo of selective segmentation models [5,6,41,44,52,53,61] from the last few years. From these, we choose to focus on three recently introduced convex models: (1) the selection model [61] based on Euclidean distance, (2) the modified selection model based on geodesic distance [59] and (3) an extension of the geodesic model for images with similar average foreground and background intensities [57]. These may be represented by

$$\min_u \left\{ \int_\Omega \mathcal{R}(u)\,\mathrm{d}\Omega + \int_\Omega \mathcal{F}u\,\mathrm{d}\Omega + \int_\Omega \mathcal{D}u\,\mathrm{d}\Omega + \alpha\int_\Omega \nu_{\varepsilon_2}(u)\,\mathrm{d}\Omega \right\}, \tag{5}$$

where $\mathcal{R}(u)$ is the regularization term, which ensures the problem has a unique solution, $\mathcal{D}:\Omega\to\mathbb{R}$ is a pre-defined distance function (to be discussed below) that helps to localize the intended object, and $\mathcal{F}:\Omega\to\mathbb{R}$ is an intensity fitting term. We note that the techniques in this paper apply to a huge array of convex segmentation models [6,15,19,22,23,32,41,42,44,48,64,70] beyond the above three.

Euclidean Model. Firstly, we mention the model of Spencer *et al.* [61] whose distance penalty is the normalized Euclidean distance from a polygon $\mathcal{P}$ formed by the points in $\mathcal{M}$. Explicitly, their distance function is $\mathcal{D} = \mathcal{E}/\|\mathcal{E}\|_\infty$, for $\mathcal{E}$ the Euclidean distance of each pixel in the image from $\mathcal{P}$. Inside $\mathcal{P}$ we set $\mathcal{D} = 0$. This is a good distance term but fails in some cases where the object is large or has a complex shape, as the Euclidean distance cannot account for this.

Geodesic Model. Extending from the Euclidean model, one may use a modified geodesic distance function $\mathcal{D}_G$ from the polygon $\mathcal{P}$ in the image segmentation model, as introduced by [59]. The initial

geodesic distance $\mathcal{D}_G^0(\boldsymbol{x})$ is obtained as the solution to

$$|\nabla \mathcal{D}_G^0(\boldsymbol{x})| = f(\boldsymbol{x}), \quad \mathcal{D}_G^0(\boldsymbol{x}^0) = 0, \quad \boldsymbol{x}^0 \in \mathcal{M}, \tag{6}$$

where $f(\boldsymbol{x}) = g(|\nabla z(\boldsymbol{x})|)$ is the edge-detector from (2). The geodesic model is given by

$$\begin{aligned}\min_{u,c_1,c_2} F_{GEO}(u,c_1,c_2) = \mu \int_\Omega g(|\nabla z(\boldsymbol{x})|)|\nabla u| \,\mathrm{d}\Omega + \theta \int_\Omega \mathcal{D}_G u \,\mathrm{d}\Omega \\ + \int_\Omega \left[\lambda_1 (z(\boldsymbol{x}) - c_1)^2 - \lambda_2 (z(\boldsymbol{x}) - c_2)^2\right] u \,\mathrm{d}\Omega + \alpha \int_\Omega \nu(u) \,\mathrm{d}\Omega,\end{aligned} \tag{7}$$

with μ, λ_1, λ_2 and θ fixed non-negative tuning parameters and α as defined for (3). To solve this model, we find the Euler-Lagrange equation. This is given by

$$\mu \nabla \cdot \left(g(|\nabla z(\boldsymbol{x})|) \frac{\nabla u}{|\nabla u|_{\varepsilon_1}} \right) - \theta \mathcal{D}_G - \left[\lambda_1 (z(\boldsymbol{x}) - c_1)^2 - \lambda_2 (z(\boldsymbol{x}) - c_2)^2\right] - \alpha \nu'_{\varepsilon_2}(u) = 0, \tag{8}$$

with Neumann boundary condition $\partial u / \partial \boldsymbol{n} = 0$ for $\boldsymbol{n}$ the unit outward normal and ε_1 and ε_2 small positive values as in (4)

Similar Foreground-Background (SFB) Model. The above models were extended by Roberts and Spencer [57] to address the failure of the intensity fitting terms in selective segmentation when the average intensities of the foreground and background are similar. They propose changing the fitting term $\mathcal{F}$ to

$$\mathcal{F} = (z - c_1)^2 - \tilde{f}(z), \quad \tilde{f}(z) = \begin{cases} 1 + \dfrac{z(\boldsymbol{x}) - c_1}{\gamma_1}, & c_1 - \gamma_1 \le z(\boldsymbol{x}) \le c_1, \\ 1 - \dfrac{z(\boldsymbol{x}) - c_1}{\gamma_2}, & c_1 < z(\boldsymbol{x}) \le c_1 + \gamma_2, \\ 0, & \textit{else}, \end{cases} \tag{9}$$

from the Chan-Vese terms used in earlier models. Here the function $\tilde{f}(z)$ is asymmetric and piecewise linear. It takes value 1 at the value c_1 and decreases to zero away from c_1 at a defined rate. The values γ_1 and γ_2 give thresholds on the distance above and below c_1 where non-zero values are taken. These values γ_1 and γ_2 are automatically determined, see [57] for details. This new fitting term is more appropriate for selective segmentations of images where $c_1 \approx c_2$. This occurs commonly when we segment the lung in CT scans, as the lung has low intensity (so c_1 is small) but the pixels outside of the patient are all of low intensity too, therefore c_2 is very similar to c_1. In this case, using the Chan-Vese intensity fitting terms in the segmentation model normally fails, however the fitting term in (9) is robust and allows accurate segmentations in cases with $c_1 \approx c_2$.

3. A unified optimization model and difficulties in fast solution

We now consider a generic variational model and a unified image segmentation model (which is a special case of the generic model). We will give the key issues associated with developing a fast non-linear multigrid full approximation scheme solver for the unified model specifically, but which also apply to the generic model.

A Generic Variational Model. Although our application interest is in image segmentation models, the techniques discussed in this paper can be in principle applied to any model of the form:

$$\min_{u \in [0,1]} \left\{ \mu \int_\Omega \mathcal{R}(u) \,\mathrm{d}\Omega + \lambda \int_\Omega \mathcal{G}(u) \,\mathrm{d}\Omega \right\} \tag{10}$$

e.g. from modeling an inverse problem. Here $\mathcal{R}(u)$ represents a regulariser term (generally non-linear) and $\mathcal{G}(u)$ is a data fidelity term. There are a huge number of variational models which fall

into this class, some examples being Badshah *et al.* [6], Klodt *et al.* [41], Nguyen *et al.* [48], Brown *et al.* [18], Goldluecke and Cremers [29], Cremers *et al.* [25], Ochs and Brox [49], Tai *et al.* [62] and Liu *et al.* [44]. Therefore, our approach can potentially be applied to a large set of existing models.

Application to Image Segmentation Models. Below we consider a specific form of (10) by fixing the regularization term as $\mathcal{R}(u) = g(|\nabla z(\boldsymbol{x})|)|\nabla u|$ the weighted total variation (TV) regulariser [51], with μ a fixed non-negative tuning parameter and g an edge detector as defined for (2). The data fidelity term $\mathcal{G}(u)$ is the same as in the generic model. The key difference from the generic model to the unified segmentation model is the fixing of the regulariser to one used throughout image segmentation (edge weighted TV) and the inclusion of the convex-relaxation penalty term $\nu_{\varepsilon_2}(u)$ to enforce the constraint that $u \in [0, 1]$. We therefore propose to study the unconstrained convex unified segmentation model

$$\min_u \left\{ \mu \int_\Omega g(|\nabla z(\boldsymbol{x})|)|\nabla u| \, \mathrm{d}\Omega + \lambda \int_\Omega \mathcal{F}u \, \mathrm{d}\Omega + \theta \int_\Omega \mathcal{D}u \, \mathrm{d}\Omega + \alpha \int_\Omega \nu_{\varepsilon_2}(u) \, \mathrm{d}\Omega \right\} \tag{11}$$

which we shall denote as model **A0**. In the interests of completion, we show how the models discussed earlier fit into this class:

- *Convex CV* : $\mathcal{F} = \lambda_1(z(\boldsymbol{x}) - c_1)^2 - \lambda_2(z(\boldsymbol{x}) - c_2)^2$, $\mathcal{D} = 0$.
- *Euclidean Model*: $\mathcal{F} = \lambda_1(z(\boldsymbol{x}) - c_1)^2 - \lambda_2(z(\boldsymbol{x}) - c_2)^2$, $\mathcal{D} =$ Euclidean Distance.
- *Geodesic Model*: $\mathcal{F} = \lambda_1(z(\boldsymbol{x}) - c_1)^2 - \lambda_2(z(\boldsymbol{x}) - c_2)^2$, $\mathcal{D} =$ Geodesic Distance.
- *SFB Model*: $\mathcal{F} = (z - c_1)^2 - \tilde{f}(z)$, $\mathcal{D} =$ Geodesic Distance.

The corresponding Euler-Lagrange equation from minimizing the convex imaging functional in (11) is given by

$$\mu \nabla \cdot \left(g(|\nabla z(\boldsymbol{x})|) \frac{\nabla u}{|\nabla u|_{\varepsilon_1}} \right) - \lambda \mathcal{F} - \theta \mathcal{D} - \alpha \nu'_{\varepsilon_2}(u) = 0, \tag{12}$$

with Neumann boundary condition $\partial u / \partial \boldsymbol{n} = 0$ for $\boldsymbol{n}$ the unit outward normal and ε_1 and ε_2 small positive values as in (4). Our aim is to develop a fast algorithm to solve the non-linear PDE (12). However this is a challenging task.

Discretisation of the PDE. Using standard second order finite differences, the PDE (12) is discretized to the form

$$\begin{aligned} &\mu(G_{i+1/2,j}(u_{i+1,j} - u_{i,j}) + G_{i-1/2,j}(u_{i-1,j} - u_{i,j}) + G_{i,j+1/2}(u_{i,j+1} - u_{i,j}) \\ &\quad + G_{i,j-1/2}(u_{i,j-1} - u_{i,j})) - \lambda \mathcal{F}_{i,j} - \theta \mathcal{D}_{i,j} - \alpha \nu'_{\varepsilon_2}(u_{i,j}) = 0, \end{aligned} \tag{13}$$

where $\mathcal{F}$ and $\mathcal{D}$ are defined appropriately, $G_{i+1/2,j} = g(|\nabla z(\boldsymbol{x})|)_{i+1/2,j} / h_x^2 (|\nabla u|_{\varepsilon_1})_{i+1/2,j}$ and $G_{i-1/2,j}$, $G_{i,j+1/2}$, $G_{i,j-1/2}$ are defined similarly. We want to solve the system of N equations resulting from this discretisation using an iterative method.

Non-Linear Multigrid (NMG). Multigrid algorithms are known to be optimal for certain applications, so we consider them first. They have optimal complexity $\mathcal{O}(N)$ where N is the number of pixels in the image. Fundamentally, we seek to perform the majority of the computations on more coarsely discretized grid than the initial grid. We first must discretize the image domain Ω for a given coarse and fine grid spacing.

For image domain $\Omega = [0, 1]^2$, denote by Ω^h the discretized image domain with grid spacing h, and similarly Ω^{2h} has grid spacing $2h$.

Full Approximation Scheme. There are various non-linear multigrid algorithms, see [34,37] and references therein, we will focus on the popular Full Approximation Scheme (FAS) of Brandt [12]. The key aspects of the algorithm are:

- *Interpolation*: The operator I_{2h}^{h} transfers values from Ω^{2h} to Ω^{h}. We use bilinear interpolation [63].
- *Restriction*: The operator I_{h}^{2h} transfers values from Ω^{h} to Ω^{2h}. We use full-weighting restriction [63].
- *Coarse Grid Solution*: This is the exact solver on Ω^{2h}, which consumes the majority of the computational power. The solver should be accurate. We use Additive Operator Splitting (AOS) [65], but we may use any accurate algorithm for solving non-linear equations.
- *Smoothing*: This part of the algorithm removes high-frequency components in the error terms e^{h} and e^{2h}.
  This ensures that a coarse grid solution focuses on removing low-frequency errors, so that we obtain an accurate error approximation on the fine grid. There is some degree of choice in which smoother we use. This step tends to be the most crucial in a multigrid method.

Choice of Smoothers. Typically, when solving a non-linear problem using FAS or some other multigrid framework, the choice of smoother is incredibly important and must be made with care. This is due principally to the potential for discontinuous coefficients in the discretized PDE or anisotropies in the PDE. In this section we will briefly review three smoothers used in the linear and non-linear multigrid frameworks. Two of the most commonly used smoothing schemes are lexicographic Gauss-Seidel (GSLEX) and line Gauss-Seidel (GSLINE), these perform well usually for a PDE with continuous coefficients. We also consider a non-standard smoother (GSHYBRID) introduced in an earlier work [58]. GSHYBRID is demonstrated to overcome the poor performance of GSLEX and GSLINE for PDEs with discontinuous coefficients by focusing on the pixels at discontinuities and performing non-standard iterations of a smoothing scheme.

GSLEX. This is simply the pixel-by-pixel update of the values $u_{i,j}$ in a sequential manner for $(i, j) \in \Omega$.

GSLINE. With this smoother we collectively update a line of pixels at a time. This type of smoother is used for anisotropic PDEs, where smoothing is preferred in one direction along the grid [63]. We can rearrange the discretized PDE by moving all $u_{\cdot,j}$ terms which are in the same column j to the left hand side to derive an iteration method.

GSHYBRID. This is a non-standard smoother. See [56,58]. It is designed for PDEs with non-smooth coefficients, i.e. PDEs such as $\nabla \cdot (K\nabla u) = f$, where K is discontinuous. Briefly, the smoother isolates those pixels which have discontinuous coefficients and performs partial line smoothing operations locally (for the identified pixels only). It performs GSLEX on all pixels without discontinuous coefficients. This smoother is found to be theoretically and experimentally effective in an FAS algorithm to solve PDEs with discontinuous coefficients.

Local Fourier Analysis. Local Fourier Analysis (LFA) gives a quantitative measure for the effectiveness of a smoother [12,24,63]. The technique measures the largest amplification factor that the smoother has on high-frequency errors. Using LFA, we may compute the smoothing rates and show that the above mentioned smoothers for the discretized segmentation models are not suitable for FAS. Hence NMG methods for (11) do not converge in a reliable way.

3.1. Key problems for convergence of NMG

The following important remark explains why we cannot implement a multigrid scheme for model **A0** (11) directly, which leads us to consider reformulations of the model.

Remark 3.1: The solution u of the Euler-Lagrange equation for the general convex model we consider, given by (12), is binary almost everywhere [23]. Therefore, the majority of pixels have values in the interval $I_\delta = [-\delta, \delta] \cup [1-\delta, 1+\delta]$ for arbitrary small δ. However, it is precisely for values in I_δ that for small changes in the value of $u_{i,j}$ we have significant jumps in the value of $\nu'_{\varepsilon_2}(u_{i,j})$ (see [23,61] for details). This leads to instability in the fixed point smoothing scheme used in the FAS algorithm.

Therefore, we find that we simply cannot implement FAS directly for model **A0**, given by (11), as the instability in the smoothing scheme leads to a non-converging multigrid algorithm.

To motivate our solution, we identify four key issues in implementing the above FAS algorithm directly on Equation (13). We detail each below, some can be seen directly from the equation and the rest are noted through experimentation.

Problem 3.1 (Jumps in $v'_{\varepsilon_2}(u)$): *The value of ε_2 influences convergence and the algorithm will simply not converge unless ε_2 is large (i.e. $\varepsilon_2 \geq 1$). This is due to the jumps in the function $v'_{\varepsilon_2}(u_{i,j})$ around* 0 *and* 1 *and the smoothing iterative scheme is unstable. So as u almost binary, if ε_2 is too small then a small change in the value of $u_{i,j}$ (being around* 0 *or* 1*) can lead to a significant change in $v'_{\varepsilon_2}(u_{i,j})$.*

Problem 3.2 (Convergence dependent on ε_1): *The quantity ε_1 should be a very small number, its only purpose is to ensure that a singularity isn't obtained in the PDE* (12) *when $|\nabla u| = 0$. However we, and other authors* [2,17,21,28]*, find that the convergence rate of the overall FAS scheme is heavily dependent on the value of ε_1. It also changes dependent on the image size and values $\varepsilon > 10$ can be necessary to ensure convergence. This is not true to the model* (11) *however as the parameter should be almost zero.*

Problem 3.3 (Inaccurate approximations of g and $|\nabla u|$ at half points): *The discretisation* (13) *requires us to approximate accurately the values $g_{i+1/2,j}$, $(|\nabla u|_{\varepsilon_1})_{i+1/2,j}$ etc. However, the edge detector g is generally non-smooth, and the function u is binary almost everywhere so it is highly non-smooth. Hence, these approximations at half pixels can be very inaccurate.*

Problem 3.4 (Inaccurate approximation of the non-linear operator): *The first term in the PDE* (12) *is very non-linear and, in the FAS algorithm (see, e.g.* [63]*), we require the accurate computation of the non-linear operator N everywhere. However, the approximation of this operator in* (13) *around the interfaces (jumps) in g and u is inaccurate (as in Problem* 3.2*). Therefore we obtain an incorrect error correction term and inaccurate solution overall.*

Problems 3.1 and 3.2 are solved by setting the respective values ε_2 and ε_1 large, however we would like to develop a framework in which the solver does not depend on these values for convergence. In the case of Problems 3.3 and 3.4, we could use the Immersed Interface techniques of [1,43] to obtain more accurate solutions around the interfaces. However, these are very non-trivial to implement and we are looking for a simpler solution.

Below we consider how to reformulate model **A0** in order to apply multigrid methods. Before that, we mention some related works.

3.2. Related fast iterative methods

Recently, a few papers have considered fast solvers for segmentation models. We briefly review three papers related to this one.

D'Ambra-Tartaglione [26]. The work in [26] proposed a linear multigrid algorithm for solving a coupled PDE system stemming from the Ambrosio-Tortorelli segmentation model. This model, not of the form (5), approximates the length term (related to Γ) by two Γ-convergence terms which lead to Laplacian type operators without the non-linearity as in (5). They choose to use a smoother similar to GSLEX. The CPU times (T) in numerical experiments of [26] show that their algorithm is sub-optimal (since the CPU ratio T^h/T^{2h} is more than 4). It is potentially interesting to employ alternative smoothers and to add LFA to guide such choices, and also to extend [26] further to segment objects selectively.

Zhang-Chen-Gould [69]. The algorithm given in [69] is a more traditional gradient descent method for a local banded variant of the Badshah-Chen model [5] which only considers pixels in a small band

around the interface where $\phi = 0$. This banding idea drastically reduces the number of unknowns to be updated in each iteration. To ensure the underlying level function ϕ remains as a signed-distance function, a multigrid method is employed to solve the equation $\nabla\phi = 1$ in a domain that includes the boundary Γ. When such Γ is of smooth rounded shape (e.g. the liver), the method works well but it performs poorly for other irregular or large shapes where the solution domain may be close to, or coincide with Ω.

Jumaat-Chen [39]. The work by [39] proposed a multilevel algorithm for the Rada-Chen [53] non-convex model and its modified variant using the band idea. The algorithm solves the discretized (non-linear and smooth) using a discretize then optimize approach. That algorithm may be applicable to our variational models, however we must smooth the functional first. Another potential drawback is that the algorithm cycles through pre-set search directions and it is not yet possible to accelerate it further or select part of these directions to save time.

4. Reformulation of the non-linear model (eqn11)

We now consider how to reformulate the unconstrained convex model (11) so that resulting PDEs are much simpler than (12). Two ideas respectively from Bresson *et al.* [15] and Goldstein *et al.* [30] are first employed to this reformulation purpose. As it turns out, none can solve the problem satisfactorily. We then combine the two ideas in order to solve all of the four problems described in Section 3.1. This does not come for free, and we have to solve four subproblems, however three of these have closed form solutions and the other is linear. We also introduce two new parameters, however we will see later that these are fixed for most experiments and the results are robust to changes in the new parameters.

Specifically, we propose two reformulations of the model **A0**. The first reformulated model **A1** still requires us to solve a non-linear problem, whereas model **A2** only requires solving a linear problem. **A2** is our recommended choice. For both **A1** and **A2** we are able to develop convergent multigrid methods.

4.1. The first Algorithm A1

Using an idea from Bresson *et al.* [15] (for a different minimization problem), we reformulate the model (11) to remove the penalty term $\nu_{\varepsilon_2}(u)$ from the formulation. This is done by introducing a new variable v and minimizing over both the u and v variables separately. Initially, (11) is reformulated to the equivalent

$$\min_{u,v}\left\{\mu\int_\Omega g(|\nabla z(\boldsymbol{x})|)|\nabla u|\,\mathrm{d}\Omega + \lambda\int_\Omega \mathcal{F}v\,\mathrm{d}\Omega + \theta\int_\Omega \mathcal{D}v\,\mathrm{d}\Omega + \alpha\int_\Omega \nu_{\varepsilon_2}(v)\,\mathrm{d}\Omega + \frac{\theta_B}{2}\|u-v\|_{L^2}^2\right\}, \tag{14}$$

and we minimize over u and v alternately where θ_B is a fixed non-negative tuning parameter, typically taking a large value to enforce u and v being similar. Notice that if we assume $u = v$ this reformulation reduces back to (11).

u minimization. We first minimize over u, solving the minimization problem

$$\min_u\left\{\mu\int_\Omega g(|\nabla z(\boldsymbol{x})|)|\nabla u|\,\mathrm{d}\Omega + \frac{\theta_B}{2}\|u-v\|_{L^2}^2\right\}. \tag{15}$$

Minimisation of (15) with respect to u results in the non-linear PDE

$$\mu\nabla\cdot\left(g(|\nabla z(\boldsymbol{x})|)\frac{\nabla u}{|\nabla u|_{\varepsilon_1}}\right) - \theta_B\left(u - v^{(k)}\right) = 0, \tag{16}$$

assuming Neumann boundary condition $\partial u/\partial \boldsymbol{n} = 0$ for $\boldsymbol{n}$ the unit outward normal. This reformulation simply attempts to solve Problem 3.3 described in Section 3.1. Notice that this is the

Euler-Lagrange equation for weighted TV denoising, a problem which has been tackled before in the literature [20,51]. But a reliable NMG algorithm is still missing.

v minimization. Then we minimize over v, solving the minimization problem

$$\min_{v} \left\{ \lambda \int_{\Omega} \mathcal{F} v \, \mathrm{d}\Omega + \theta \int_{\Omega} \mathcal{D} v \, \mathrm{d}\Omega + \alpha \int_{\Omega} v_{\varepsilon_2}(v) \, \mathrm{d}\Omega + \frac{\theta_B}{2} \|u - v\|_{L^2}^2 \right\}, \tag{17}$$

which has a closed form solution [15]

$$v^{(k+1)} = v = \min \left\{ \max \left\{ u - \frac{\lambda \mathcal{F} + \theta \mathcal{D}}{\theta_B}, 0 \right\}, 1 \right\}. \tag{18}$$

We note that, as we wanted, we have removed v'_{ε_2} from the formulation. In summary, **A1** is given by the following algorithm:

(1) Solve PDE (16) for u.
(2) Update v by (18).
(3) If $\|u^{(k+1)} - u^{(k)}\| <$ tolerance and $\|v^{(k+1)} - v^{(k)}\| <$ tolerance, stop, else return to 1.

The outstanding problem with this algorithm is that it relies on another reliable NMG algorithm for (16).

4.2. Further splitting of A1

To remove non-linearity in the TV term in (11), we now consider the idea of Split-Bregman iteration, as introduced by Goldstein *et al.* [30] for image restoration. This is a fast and accurate iterative solver for models with TV like functionals. Here we replace the weighted TV term in the minimization with a new variable $\boldsymbol{d}$. This splitting does not rely on a regularization parameter in the PDE and leads therefore to reduction of non-linearity. We can rewrite the minimization problem (11) as

$$\min_{u,\boldsymbol{d}} \left\{ \mu \int_{\Omega} |\boldsymbol{d}|_g \, \mathrm{d}\Omega + \lambda \int_{\Omega} \mathcal{F} u \, \mathrm{d}\Omega + \theta \int_{\Omega} \mathcal{D} u \, \mathrm{d}\Omega + \alpha \int_{\Omega} v_{\varepsilon_2}(u) \, \mathrm{d}\Omega + \frac{\lambda_B}{2} \|\boldsymbol{d} - \nabla u - b\|_{L^2}^2 \right\}, \tag{19}$$

where $|\boldsymbol{d}|_g = g(|\nabla z(\boldsymbol{x})|)|\nabla u|$ and λ_B is a fixed non-negative tuning parameter. Note that b is the Bregman update which has a simple update formula. We minimize over u and $\boldsymbol{d}$ in (19):

u minimization. This minimization is given by

$$\min_{u} \left\{ \lambda \int_{\Omega} \mathcal{F} u \, \mathrm{d}\Omega + \theta \int_{\Omega} \mathcal{D} u \, \mathrm{d}\Omega + \alpha \int_{\Omega} v_{\varepsilon_2}(u) \, \mathrm{d}\Omega + \frac{\lambda_B}{2} \|\boldsymbol{d} - \nabla u - b\|_{L^2}^2 \right\}. \tag{20}$$

***d** minimization*. This minimization is given by

$$\min_{\boldsymbol{d}} \left\{ \mu \int_{\Omega} |\boldsymbol{d}|_g \, \mathrm{d}\Omega + \frac{\lambda_B}{2} \|\boldsymbol{d} - \nabla u - b\|_{L^2}^2 \right\} \quad \Rightarrow \quad \boldsymbol{d} = \mathrm{shrink}\left(\nabla u + b, \frac{\mu g(|\nabla z(\boldsymbol{x})|)}{\lambda_B} \right), \tag{21}$$

where $\mathrm{shrink}(u, a) = \mathrm{sgn}(u) \max\{|u| - a, 0\}$ from [30] and we update b using the formula $b^{(k+1)} = b^{(k)} + \nabla u^{(k+1)} - \boldsymbol{d}^{(k+1)}$. See [30] for the proof that these closed form solutions solve the minimization problem for $\boldsymbol{d}$. The essential step in the above Bregman iterations is the solution of (20) which is unfortunately still non-linear, not amenable to fast NMG solution. Further refinement is required.

4.3. The second and recommended Algorithm A2

Our recommended algorithm **A2** tries to tackle the non-linearity in u minimization, given by (15) in **A1**, by using the Split-Bregman idea of Section 4.3 to solve this subproblem. This incidentally addresses Problems 3.2–3.4 from Section 3.1. Specifically we use the ideas, as with (20), to replace the weighted TV term in (15) with new variables $\boldsymbol{d}$ and b and split the minimization into a total of 4 problems (for u, $\boldsymbol{d}$, b and v). Three subproblems ($\boldsymbol{d}$, b and v) have a closed form solution and the other problem is linear (as we will see in (22)). We reformulate the minimization problem (15) to

$$\min_{u,\boldsymbol{d}} \left\{ \mu \int_\Omega |\boldsymbol{d}|_g \, \mathrm{d}\Omega + \frac{\theta_B}{2} \|u - v\|_{L^2}^2 + \frac{\lambda_B}{2} \|\boldsymbol{d} - \nabla u - b\|_{L^2}^2 \right\}. \tag{22}$$

using Bregman splitting where θ_B and λ_B are fixed non-negative parameters. At convergence, we aim to have $\boldsymbol{d} = |\nabla u|$ and therefore b is near zero, see [30] for proof that this reformulation is equivalent to (15). Therefore this splits into three subproblems, the u and $\boldsymbol{d}$ minimization and b update, which along with the v update (18) gives us the following four subproblems.

u minimization. This is obtained as the solution of

$$u^{(k+1)} = \arg\min_u \left\{ \frac{\theta_B}{2} \|u - v^{(k)}\|_{L^2}^2 + \frac{\lambda_B}{2} \|\boldsymbol{d}^{(k)} - \nabla u - b^{(k)}\|_{L^2}^2 \right\},$$

and the minimizer is the solution of

$$-\lambda_B \Delta u + \theta_B u = \theta_B v^{(k)} - \lambda_B \nabla \cdot \left(\boldsymbol{d}^{(k)} - b^{(k)} \right). \tag{23}$$

with Neumann boundary conditions. This linear PDE is solved by any convergent algorithm, e.g. AOS [65] or linear multigrid [63].

d *minimization*.

$$\boldsymbol{d}^{(k+1)} = \text{shrink}\left(\nabla u^{(k+1)} + b^{(k)}, \frac{\mu g(|\nabla z(\boldsymbol{x})|)}{\lambda_B} \right). \tag{24}$$

b update.

$$b^{(k+1)} = b^{(k)} + \nabla u^{(k+1)} - \boldsymbol{d}^{(k+1)}. \tag{25}$$

v minimization. As for **A1**, we set

$$v^{(k+1)} = \min\left\{ \max\left\{ u^{(k+1)} - \frac{\lambda \mathcal{F} + \theta \mathcal{D}}{\theta_B}, 0 \right\}, 1 \right\}. \tag{26}$$

Remark 4.1: Note that with this 4 step minimization we have solved the four problems detailed in Section 3.1. Although this requires solving three more problems than we would traditionally, and introduces two new parameters. Thankfully the problems all have closed form solutions which can be computed very efficiently.

In summary, **A2** is given by the following algorithm:

(1) Solve PDE (23) for u.
(2) Update $\boldsymbol{d}$ by (24).
(3) Update b by (25).
(4) Update v by (26).
(5) If $\|u^{(k+1)} - u^{(k)}\| <$ tolerance, $\|\boldsymbol{d}^{(k+1)} - \boldsymbol{d}^{(k)}\| <$ tolerance, $\|b^{(k+1)} - b^{(k)}\| <$ tolerance and $\|v^{(k+1)} - v^{(k)}\| <$ tolerance, stop, else return to 1.

Remark 4.2: In [9] the authors prove that the multigrid algorithm converges for the class of PDEs of the form $-\nabla(a\nabla u) + bu = f$, where $a \in C^1(\bar{\Omega}), b \in C(\bar{\Omega}), a(\boldsymbol{x}), b(\boldsymbol{x}) \geq 0$ and zero Dirichlet boundary conditions. Clearly, the PDE (23), which we are solving as part of **A2** belongs to this class of PDEs. However, we have zero Neumann boundary conditions, rather than the zero Dirichlet boundary conditions in the paper. It would be an interesting piece of work to determine if the proof applies in the case of zero Neumann boundary conditions also as this would give an explicit proof that **A2** converges. We will see later that, experimentally, there is always convergence.

5. Solution algorithms on a single level and multilevels

In the previous section we discussed several PDEs whose solution is the minimizer for a given energy functional. In this section, we will discuss the two solution frameworks; single level and multilevel. We begin in Section 5.1 by discussing the single level solver for the models **A0**, **A1** and **A2**. Then in Sections 5.2–5.4 we will discuss the multilevel solutions for **A1** and **A2**. We do not consider the multilevel solution for **A0** as the algorithm simply does not converge for any parameters.

5.1. Single level algorithm – additive operator splitting for A0, A1 and A2

Additive Operator Splitting (AOS) [31,45,65] is a widely used method [3–6,53,61] for solving equations of the form $\partial u/\partial t = \mu\nabla \cdot (K(u)\nabla u) - f$. It allows us to split the two-dimensional problem into two one-dimensional problems, which we solve separately and then combine the results. Each one-dimensional problem gives rise to a tridiagonal system of equations which can be solved efficiently, hence AOS is a very efficient method for solving diffusion-like equations. AOS is a semi-implicit method and permits far larger time-steps than the corresponding explicit schemes would. Hence, AOS is more stable than an explicit method [65]. For our case where the source term f depends on u and the standard AOS must be modified using the idea of [61] (as shown below).

AOS Scheme for **A0**. Rewriting the PDE (12), resulting from minimization of model **A0**, as $\partial u/\partial t = \mu(\partial_x(G(u)\partial_x u) + \partial_y(G(u)\partial_y u)) - f(u)$ where $f(u) = \lambda\mathcal{F} + \theta\mathcal{D} + \alpha v'_{\varepsilon_2}(u)$, the discretized form of PDE (12) can be solved by AOS:

$$u^{(k+1)} = \frac{1}{2}\sum_{\ell=1}^{2}\left(I - 2\tau\mu A_\ell\left(u^{(k)}\right)\right)^{-1}\left(u^{(k)} + \tau f\left(u^{(k)}\right)\right), \tag{27}$$

where τ is the time-step, $A_1(u) = \partial_x(G(u)\partial_x)$ and $A_2(u) = \partial_y(G(u)\partial_y)$. For notational convenience we write $G = G(u)$. The matrix $A_1(u)$ can be obtained as follows

$$\begin{aligned}(A_1(u^{(k)})u^{(k+1)})_{i,j} &= (\partial_x(G\partial_x u^{(k+1)}))_{i,j}\\ &= \left(\frac{G_{i+1/2,j}}{h_x^2}\right)u_{i+1,j}^{(k+1)} + \left(\frac{G_{i-1/2,j}}{h_x^2}\right)u_{i-1,j}^{(k+1)} - \left(\frac{G_{i+1/2,j}+G_{i-1/2,j}}{h_x^2}\right)u_{i,j}^{(k+1)},\end{aligned}$$

where h_x is the grid spacing in the x-direction and, similar to [53,61], for the half points in G we take the average of the surrounding pixels, e.g. $G_{i+1/2,j} = (G_{i+1,j} + G_{i,j})/2$. Since f depends on u via $v'_{\varepsilon_2}(u)$ in (11), following [61], we consider the Taylor expansion of $v'_{\varepsilon_2}(u)$ around $u = 0$: $v_\varepsilon(u) = a_0 + b_0 u + \mathcal{O}(u^2)$ and around $u = 1$: $v_\varepsilon(u) = a_1 + b_1 u + \mathcal{O}(u^2)$. As $b_0 = b_1$, we denote the linear term as b from now on. Define the interval in which $v'_{\varepsilon_2}(u)$ jumps as $I_\zeta := [0-\zeta, 0+\zeta] \cup [1-\zeta, 1+\zeta]$, and the function $\tilde{b}_{i,j}^{(k)} = b$ if $u_{i,j}^{(k)} \in I_\zeta$ (and set $\tilde{b}_{i,j}^{(k)} = 0$ elsewhere. Then as with [61], we have the modified AOS below:

$$u^{(k+1)} = \frac{1}{2}\sum_{\ell=1}^{2}\left(I - 2\tau\mu(I + \tilde{B}^{(k)})^{-1}A_\ell\left(u^{(k)}\right)\right)^{-1}\left(u^{(k)} + \tau\left(I + \tilde{B}^{(k)}\right)^{-1} f\left(u^{(k)}\right)\right), \tag{28}$$

which does satisfy the convergence conditions of Weickert [65].

AOS Scheme for **A1**. Rewriting PDE (16), which results from the u minimization of **A1**, as $\partial u/\partial t = \mu(\partial_x(K(u)\partial_x u) + \partial_y(K(u)\partial_y u)) - f(u)$ where $f(u) = \theta_B(u - v)$, the PDE (16) can be similarly solved by

$$u^{(k+1)} = \frac{1}{2}\sum_{\ell=1}^{2}\left(I - 2\tau\mu B_\ell\left(u^{(k)}\right)\right)^{-1}\left(u^{(k)} - \tau f(u^{(k)})\right) \tag{29}$$

where τ is the time-step, $B_1(u) = \partial_x(K(u)\partial_x)$ and $B_2(u) = \partial_y(K(u)\partial_y)$. For notational convenience we write $K = K(u)$. The matrix $B_1(u)$ can be obtained as follows

$$\begin{aligned}(B_1(u^{(k)})u^{(k+1)})_{i,j} &= (\partial_x(K\partial_x u^{(k+1)}))_{i,j}\\ &= \left(\frac{K_{i+1/2,j}}{h_x^2}\right)u_{i+1,j}^{(k+1)} + \left(\frac{K_{i-1/2,j}}{h_x^2}\right)u_{i-1,j}^{(k+1)} - \left(\frac{K_{i+1/2,j}+K_{i-1/2,j}}{h_x^2}\right)u_{i,j}^{(k+1)},\end{aligned}$$

where h_x is the grid spacing in the x-direction. Notice that that $B_\ell(u^{(k)})$ is tridiagonal for $\ell = 1, 2$ and the above iterations can be performed very efficiently using the Thomas algorithm.

AOS Scheme for **A2**. Finally, rewriting PDE (23), which results from the u minimization of **A2**, as $\partial u/\partial t = \lambda_B(\partial_{xx}u + \partial_{yy}u) - f(u)$ where $f(u) = \lambda_B\nabla\cdot(\boldsymbol{d} - b) + \theta_B(u - v)$, the PDE (23) is solved by the AOS in the form:

$$u^{(k+1)} = \frac{1}{2}\sum_{\ell=1}^{2}\left(I - 2\tau\lambda_B C_\ell\left(u^{(k)}\right)\right)^{-1}\left(u^{(k)} - \tau f\left(u^{(k)}\right)\right), \tag{30}$$

where $C_1(u^{(k)}) = \partial_{xx}u$ and $C_2(u^{(k)}) = \partial_{yy}u$. When $\ell = 1$, the matrix C_1 can be obtained as follows

$$\left(C_1 u^{(k+1)}\right)_{i,j} = \left(\partial_{xx}u^{(k+1)}\right)_{i,j} = \left(\frac{1}{h_x^2}\right)u_{i+1,j}^{(k+1)} + \left(\frac{1}{h_x^2}\right)u_{i-1,j}^{(k+1)} - \left(\frac{2}{h_x^2}\right)u_{i,j}^{(k+1)},$$

where h_x is the grid spacing in the x-direction. We treat $\ell = 2$ similarly.

5.2. Multilevel algorithm for A1 and A2

The multilevel framework permits us to solve the discretized forms of these PDEs quickly, as the multigrid method has optimal complexity $\mathcal{O}(N)$ (in certain situations) [63]. We will first introduce Linear Multigrid (LMG), which is the precursor to the non-linear multigrid (NMG) FAS algorithm introduced earlier. We use LMG to solve linear PDEs and NMG to solve non-linear PDEs.

As far as implementation is concerned, NMG and LMG are almost identical except that the error correction quantity e^h for NMG can only be obtained by taking the difference $\mathrm{e}^h = w^h - \bar{w}^h$ of two approximations $w^h, \bar{w}^h$ while the LMG can compute the error correction e^h directly. That is, the difference lies in storage (or approximation). More precisely, denote a discretized linear PDE and a non-linear PDE respectively by $\mathcal{L}^h u = f^h$ and $\mathcal{N}^h u = f^h$. If we have an approximation u^h to u on Ω^h, the residual is denoted by $r^h = f^h - \mathcal{L}^h u^h$ or for NMG $r^h = f^h - \mathcal{N}^h u^h$. Then the error e^h in our approximation is found respectively by the relation

$$\begin{aligned}&\text{LMG:}\quad \mathcal{L}^h\,\mathrm{e}^h = r^h,\\ &\text{NMG:}\quad \mathcal{N}^h w^h = r^h + \mathcal{N}^h\bar{w}^h,\quad \mathrm{e}^h = w^h - \bar{w}^h.\end{aligned}$$

We update our approximation by $u = u^h + \mathrm{e}^h$. See [24,34,63].

5.3. Linear multigrid algorithm

For completeness, we show the linear multigrid algorithm in Algorithm 5.1. We introduce the parameter η which permits a general η-cycling, i.e. if $\eta = 1$ this is a simple V-cycle, if $\eta = 2$ we have a W-cycle, etc, see [33,63]. The most important component, *smoothing*, will be discussed shortly. For the other two components, *restriction* I_h^{2h} and *interpolation* I_{2h}^h, used to transfer information between the grids Ω^h, Ω^{2h}, ..., we use full-weighting restriction and bilinear interpolation [27,63].

Algorithm 5.1 Linear Multigrid Scheme, $u^h \leftarrow LMG(u^h, N^h, f^h, \nu_1, \nu_2, Smoother, level, max\_level, \eta)$

Pre-smoothing: Perform ν_1 iterations of the smoother: $u^h \leftarrow Smoother(u^h, f^h)$.

Coarse grid correction: Compute the residual: $r^h = f^h - \mathcal{L}^h u^h$.

Transfer the residual to Ω^{2h} by restriction: $r^{2h} = I_h^{2h} r^h$.

if $level = max\_level$ **then**

Compute: $\mathrm{e}^{2h} = [\mathcal{L}^{2h}]^{-1} r^{2h}$

else

Do η cycles (steps) of

$\mathrm{e}^{2h} \leftarrow LMG(\mathbf{0}, \mathcal{L}^{2h}, r^{2h}, \nu_1, \nu_2, Smoother, level + 1, max\_level, \eta)$.

end if

Interpolation: Transfer the error to Ω^h by interpolation:$\mathrm{e}^h = I_{2h}^h \mathrm{e}^{2h}$.

Correct the fine grid approximation: $u^h = u^h + \mathrm{e}^h$.

Post-smoothing: Perform ν_2 iterations of the smoother: $u^h \leftarrow Smoother(u^h, f^h)$.

5.4. Multigrid solutions of models A1 and A2

The choice of smoother for each model is an incredibly important decision in determining how quickly the multigrid method will converge. For problems with discontinuous coefficients a non-standard smoother is recommended. We find that for **A1**, the term $\nabla \cdot (g(|\nabla z(\boldsymbol{x})|)(\nabla u/|\nabla u|)_{\varepsilon_1})$ introduces these discontinuous coefficients in its discretisation. Therefore for **A1** we will recommend the use of the hybrid smoother GSHYBRID from Section 3.1. In the case of **A2**, we must solve a linear PDE with constant coefficients, hence the discontinuities aren't present. In this case, we will show that the simple cheap GSLEX scheme from Section 3.1 is effective at damping the errors.

5.4.1. Smoother for A1

In the PDE for **A1** (16), the term $\nabla \cdot (g(|\nabla z(\boldsymbol{x})|)(\nabla u/|\nabla u|)_{\varepsilon_1})$ is discretized to give

$$\begin{aligned} &G_{i+1/2,j}\left(u_{i+1,j} - u_{i,j}\right) + G_{i-1/2,j}\left(u_{i-1,j} - u_{i,j}\right) \\ &\quad + G_{i,j+1/2}\left(u_{i,j+1} - u_{i,j}\right) + G_{i,j-1/2}\left(u_{i,j-1} - u_{i,j}\right), \end{aligned} \tag{31}$$

where $G_{i+1/2,j} = g(|\nabla z(\boldsymbol{x})|)_{i+1/2,j}/h_x^2(|\nabla u|_{\varepsilon_1})_{i+1/2,j}$, etc. One problem which results from this is that the coefficients $G_{\cdot,\cdot}$ can be highly variable and may jump significantly. Therefore, solving the PDE for $u_{i,j}$ using standard smoothers, such as GSLEX and GSLINE, we have poor smoothing rates at pixels (i, j) where the coefficients are very different from one another. This leads to slow or no convergence of FAS. This phenomenon was studied in [17,58] and new non-standard smoothers were proposed which perform different smoothing steps at the pixels with jumping coefficients. We take our lead from [58] and use the smoother GSHYBRID proposed in that paper which has much improved smoothing rate and leads to a faster converging FAS. For reasons of brevity, we will not detail the algorithm for GSHYBRID, but refer the reader to the paper [58].

5.4.2. Smoother for A2

In Section 3 we considered the GSLEX smoother. This is the simplest Gauss-Seidel smoother and, to justify its choice as the smoother for **A2**, we will consider the LFA for this smoothing scheme and show that it is effective at damping the errors for our PDE. We recall that the PDE we are solving is

$$-\lambda_B \Delta u + \theta_B u = \theta_B v^{(k)} - \lambda_B \nabla \cdot \left(\boldsymbol{d}^{(k)} - b^{(k)} \right),$$

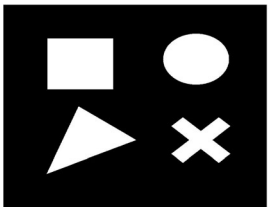
which discretises to

$$\left(\frac{4\lambda_B}{h^2} + \theta_B \right) u_{i,j} - \frac{\lambda_B}{h^2} \left(u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1} \right) = f_{i,j}, \tag{32}$$

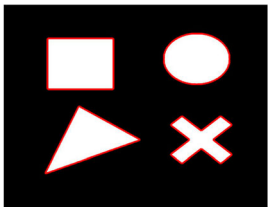
where $f_{i,j} = \theta_B v_{i,j}^{(k)} - \lambda_B \nabla \cdot (\boldsymbol{d}^{(k)} - b^{(k)})_{i,j}$ and assuming equal grid spacing in each dimension, i.e. $h = h_x = h_y$.

Lemma: *If we use the GSLEX smoother on the discretisation* (32) *of the PDE for* ***A2****, then we have global smoothing rate* $\hat{\mu} < \frac{1}{2}$.

Proof: Follows as a natural extension from the result of Brandt [13] that $\mu = \frac{1}{2}$ for Poisson's equation $-\nabla u = f$. ■



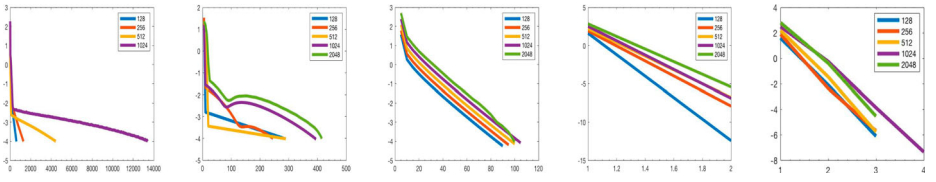
(a)



(b)

Image Size	A0 (AOS) Iters	A0 (AOS) CPU (s)	A1 (AOS) Iters	A1 (AOS) CPU (s)	A2 (AOS) Iters	A2 (AOS) CPU (s)	A1 (MG) Iters	A1 (MG) CPU (s)	A2 (MG) Iters	A2 (MG) CPU (s)	A2 (MG) Ratio Per Iter
128^2	615	11.9	285	5.3	90	1.6	2	1.3	3	0.2	-
256^2	1300	98.2	245	17.1	95	5.9	2	6.7	3	0.6	3.0
512^2	4455	1428.9	290	84.9	100	26.6	2	38.3	3	1.7	2.8
1024^2	13420	25085.1	395	547.6	105	129.7	2	205.5	4	12.2	5.4
2048^2	*	*	415	2731.4	100	684.1	2	1170.5	3	42.8	4.7

(c)



(d)

Figure 1. Test model 1 – Global segmentation for a synthetic image. (a) The given image. (b) The segmentation result. (c) Algorithm Comparisons (a * indicates the algorithm took over 24 hours). (d) Corresponding error plot with iteration number versus the base 10 logarithm of the error. From left to right are error plots for A0 (AOS), A1 (AOS), A2 (AOS), A1 (MG) and A2 (MG) respectively.

6. Numerical results

In this paper, we wish to demonstrate that our proposed multigrid algorithms for models **A1** and **A2** are efficient for solving a class of segmentation models given by (11) and especially **A2** outperforms **A1**. The variational models considered earlier, in Section 3.1, only differ in choices of regularization, data fitting and distance terms. Since the variational framework implied by form (11) is general, we believe that other models of this form beyond our tested cases can be solved similarly by our new algorithms.

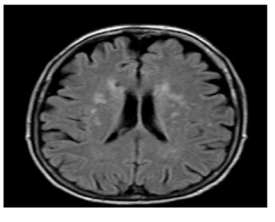
The number of unknowns that we shall consider are large, ranging from 16384 (for 128×128 images) to 4194304 $\approx 4 \times 10^6$ (for 2048×2048).

Stopping Criteria. The stopping criterion is the same for the algorithms. We stop the algorithms when $\|u^{(k+1)} - u^{(k)}\|_2 < \xi$ where we fix $\xi = 10^{-4}$ for all experiments. We note however, that in practical implementation, $\xi = 10^{-2}$ is sufficient as Γ has generally stabilized by then.

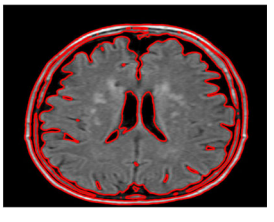
Algorithm Comparisons. We will compare the practical performance at various resolutions of the following algorithms:

- **A0 (AOS):** Single level AOS for the original model **A0** (28).
- **A1 (AOS):** Single level AOS for the first reformulation **A1** (29).
- **A2 (AOS):** Single level AOS for the second reformulation **A2** (30).
- **A1 (MG):** Multigrid for the first reformulation **A1**.
- **A2 (MG):** Multigrid for the second reformulation **A2**.

An optimal algorithm should only take $\mathcal{O}(N) = kN$ CPU times if N is the number of unknowns; however large k implies still a slower convergence rate. Here the parentheses for a model denote whether we use the single level solver **AOS** or multigrid **MG**. We omit the case of



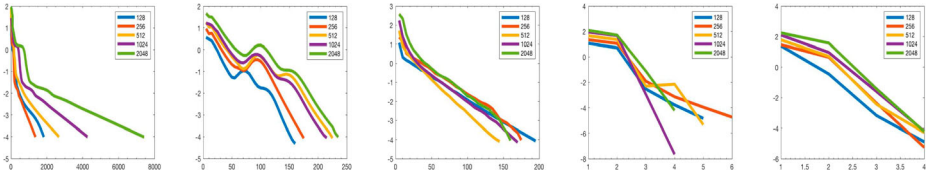
(a)



(b)

Image	A0 (AOS)		A1 (AOS)		A2 (AOS)		A1 (MG)		A2 (MG)		
Size	Iters	CPU (s)	Iters	CPU (s)	Iters	CPU (s)	Iters	CPU (s)	Iters	CPU (s)	Ratio Per Iter
128^2	1830	33.3	160	3.0	195	3.0	5	5.5	4	0.4	-
256^2	1365	103.0	175	12.3	175	12.7	6	26.8	4	0.6	1.5
512^2	2675	860.5	225	67.3	145	40.4	5	94.6	4	3.0	5.0
1024^2	4270	7253.3	215	335.1	170	211.8	4	288.7	4	15.9	5.3
2048^2	7425	72915	235	1952.7	160	1052.4	4	1520.9	4	66.4	4.2

(c)



(d)

Figure 2. Test model 1 – Global segmentation for a real (MRI) image. (a) Image. (b) Segmentation Result. (c) Algorithm Comparisons (a * indicates the algorithm took over 24 hours). (d) Corresponding error plot with iteration number versus the base 10 logarithm of the error. From left to right are error plots for A0 (AOS), A1 (AOS), A2 (AOS), A1 (MG) and A2 (MG) respectively.

a multigrid algorithm for **A0** because there does not exist such a converging MG algorithm by Remark 3.1.

Parameters. For the models **A0**, **A1** and **A2** we have many independent parameters to consider. For all models we must consider μ, θ and λ. For **A1** and **A2** we additionally have θ_B and for **A2** we also have λ_B. Firstly, we fix $\lambda_B = 1$ for **A2** in all experiments. We perform an extensive grid search over all remaining parameters for each model varying them from 10^{-3} through to 10^{10} by factors of 10. The results quoted are for those parameter values which converged in the fewest iterations and also gave Jaccard coefficient [38] over 0.95, compared to the ground truth. We test only the v-cycle, fixing $\eta = 1$ and set the edge detector parameter $\beta = 10^{-2}$. Finally, in those models where f depends on u, we update its value every 5 iterations of the AOS algorithms to permit reasonable update of u and no undue computational burden from repeated computation of f. Hence, all iteration numbers are multiples of 5.

Machine Specification. All experiments were run in MATLAB 2018b on a Linux x64_64 machine with 30GB RAM and 40 Intel Xeon Gold 5115 2.40GHz CPUs. Note that parallelization was not built into the code. Each algorithm implementation uses the same underlying codes (e.g. for gradient calculation) to permit fair comparison using CPU times.

Test Model 1: Convex Chan-Vese Global Segmentation Model (3). We give three results; a synthetic segmentation in Figure 1 and twosegmentations for real images in Figures 2 and 3. In each case we compare the time for the five algorithms to converge for five different image resolutions.

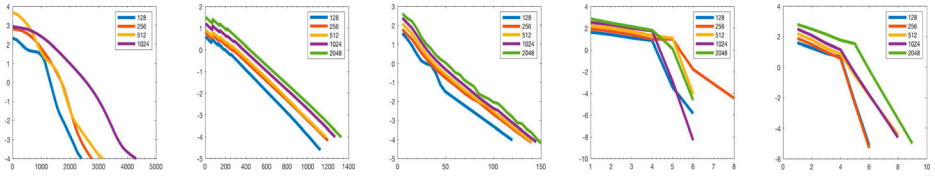
We find that the fastest algorithm is the multigrid **A2 (MG)**. For the single level solvers, we find that the timings for **A1** and **A2** are better than for the original **A0** with **A2** faster than **A1**. Finally, we find experimentally that the CPU times for each iteration of **A2 (MG)** have a linear relationship $\mathcal{O}(N)$ with the number of pixels, N, in an image, i.e. if we compare the CPU times for an image with

(a)

(b)

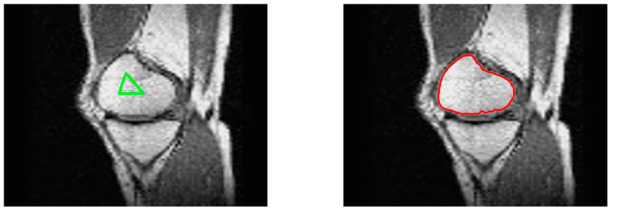
Image Size	A0 (AOS) Iters	A0 (AOS) CPU (s)	A1 (AOS) Iters	A1 (AOS) CPU (s)	A2 (AOS) Iters	A2 (AOS) CPU (s)	A1 (MG) Iters	A1 (MG) CPU (s)	A2 (MG) Iters	A2 (MG) CPU (s)	A2 (MG) Ratio Per Iter
128^2	2405	48.8	1125	16.9	120	1.8	6	6.0	6	0.4	-
256^2	2755	202.9	1200	72.5	140	8.7	8	38.9	6	0.8	2
512^2	3150	1143.2	1180	349.9	140	36.4	6	111.1	8	4.6	4.3
1024^2	4310	8399.1	1270	1724.1	145	178.4	6	754.3	8	22.3	4.8
2048^2	*	*	1330	9735.9	150	971.0	6	5563.8	9	114.8	4.6

(c)



(d)

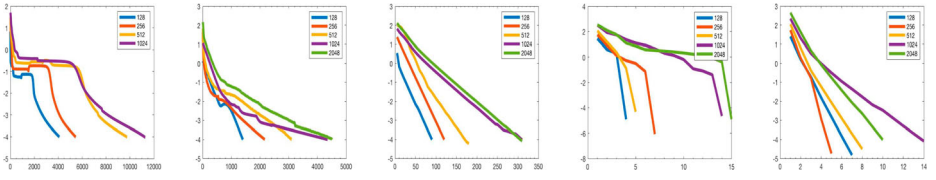
Figure 3. Test model 1 – Global segmentation for a real (standard cameraman) image. (a) Image. (b) Segmentation Result. (c) Algorithm Comparisons (a * indicates the algorithm took over 24 hours). (d) Corresponding error plot with iteration number versus the base 10 logarithm of the error. From left to right are error plots for A0 (AOS), A1 (AOS), A2 (AOS), A1 (MG) and A2 (MG) respectively.



(a) (b)

Image Size	A0 (AOS) Iters	A0 (AOS) CPU (s)	A1 (AOS) Iters	A1 (AOS) CPU (s)	A2 (AOS) Iters	A2 (AOS) CPU (s)	A1 (MG) Iters	A1 (MG) CPU (s)	A2 (MG) Iters	A2 (MG) CPU (s)	A2 (MG) Ratio Per Iter
128^2	4095	73.9	1400	24.7	90	1.3	4	3.5	7	0.4	-
256^2	5460	495.5	2165	155.5	120	7.7	7	21.6	5	0.9	3.2
512^2	9760	5030.5	3100	1164.1	180	50.1	5	47.3	8	5.1	3.5
1024^2	11280	31725.5	4355	5895.2	310	384.6	14	217.4	14	42.5	4.8
2048^2	*	*	4510	34351.5	310	2039.7	15	2314.7	10	148.8	4.9

(c)



(d)

Figure 4. Test model 2 – Selective segmentation (geodesic) for a real (MRI) image. (a) Image and Marker Set $\mathcal{M}$ (green). (b) Segmentation Result. (c) Algorithm Time Comparisons (a * indicates the algorithm took over 24 hours). (d) Corresponding error plot with iteration number versus the base 10 logarithm of the error. From left to right are error plots for A0 (AOS), A1 (AOS), A2 (AOS), A1 (MG) and A2 (MG) respectively.

$N/4 = (n/2)^2$ pixels and another with $N = n^2$ pixels, we find that there is a factor 4 increase in the computational time. Therefore, **A2 (MG)** has complexity near the optimal $\mathcal{O}(N)$. Clearly **A1 (MG)** is a few times slower than **A2 (MG)**. This (approximately) 'constant' ratio can be observed in all the subsequent tables.

Test Model 2: Geodesic Selective Segmentation Model (7). We give results for a real knee MRI scan. Here the object we are looking to segment is of high intensity compared to the background. The image and the correct segmentation result are shown in Figure 4.

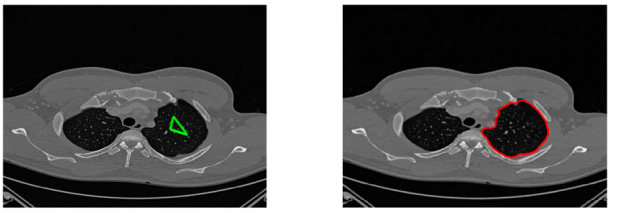
As in the previous test, we find that **A2 (MG)** gives the fastest segmentation compared to the other algorithms and **A2** outperforms **A1** for both the multigrid and the single level solver. All methods outperform **A0 (AOS)**. We also note again that **A2 (MG)** gives near optimal, i.e. $\mathcal{O}(N)$ performance.

Test Model 3: SFB Selective Segmentation Model (9). This test addresses segmenting one lung from a CT image. The challenge in this test image is that both the background and foreground have similar average intensities (where the foreground is a single segmented lung). The results of the segmentation are shown in Figure 5.

As with the previous tests we find that the multigrid implementation of **A2** performs much better than the alternative models. We also note again that **A2 (MG)** gives near optimal, i.e. $\mathcal{O}(N)$ performance.

7. Conclusions

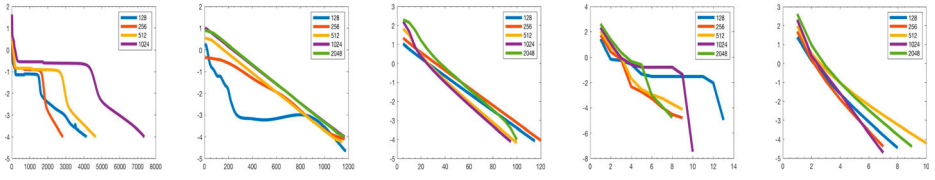
In this paper we have introduced two new reformulations for solving a class of convex non-linear image segmentation minimization problems. Typically, the PDEs which result from such minimization problems cannot be directly solved by non-linear multigrid (FAS) due to non-linearity,



(a) (b)

Image Size	A0 (AOS) Iters	A0 (AOS) CPU (s)	A1 (AOS) Iters	A1 (AOS) CPU (s)	A2 (AOS) Iters	A2 (AOS) CPU (s)	A1 (MG) Iters	A1 (MG) CPU (s)	A2 (MG) Iters	A2 (MG) CPU (s)	A2 (MG) Ratio Per Iter
128^2	4155	86.7	1180	22.1	115	1.9	13	11	8	0.3	-
256^2	2835	216.1	1170	82.1	120	7.2	9	32.2	7	1.2	4.6
512^2	4670	1758.1	1150	399.1	100	25.2	9	190.1	10	7.4	4.3
1024^2	7370	14646.6	1170	1665.1	95	119.7	10	851.1	7	26.8	5.2
2048^2	*	*	1165	8536.7	100	602.3	8	5754.2	9	152.1	4.4

(c)



(d)

Figure 5. Test model 3 – Selective segmentation (SFB) of a lung in a CT scan. (a) Image and Marker Set $\mathcal{M}$ (green) (b) Segmentation Result (c) Algorithm Time Comparisons (a * indicates the algorithm took over 24 hours). (d) Corresponding error plot with iteration number versus the base 10 logarithm of the error. From left to right are error plots for A0 (AOS), A1 (AOS), A2 (AOS), A1 (MG) and A2 (MG) respectively.

instability and no convergence. Using splitting ideas we reformulate the models and obtain two algorithms; **A1 (MG)** which uses non-linear FAS on a more stable problem, and **A2 (MG)** which solves a linear problem using linear multigrid. We have proven experimentally that the extra splittings used to obtain **A2 (MG)** lead to a faster converging algorithm. This work allows other researchers who have non-linear minimization problems to consider using splittings to reformulate their problem into a more stable non-linear problem, or even to obtain a linear problem.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

Work supported by the Engineering and Physical Sciences Research Council (UK EPSRC) grant EP/N014499/1.

References

[1] L. Adams and Z. Li, *The immersed interface/multigrid methods for interface problems*, SIAM J. Sci. Comput. 24(2) (2002), pp. 463–479.
[2] U.M. Ascher, E. Haber, and H. Huang, *On effective methods for implicit piecewise smooth surface recovery*, SIAM J. Sci. Comput. 28(1) (2006), pp. 339–358.
[3] N. Badshah and K Chen, *Multigrid method for the Chan-Vese model in variational segmentation*, Commun. Comput. Phys. 4(2) (2008), pp. 294–316.
[4] N. Badshah and K. Chen, *On two multigrid algorithms for modeling variational multiphase image segmentation*, IEEE Trans. Image Process. 18(5) (2009), pp. 1097–1106.
[5] N. Badshah and K. Chen, *Image selective segmentation under geometrical constraints using an active contour approach*, Commun. Comput. Phys. 7(4) (2010), pp. 759–778.

[6] N. Badshah, K. Chen, H. Ali, and G. Murtaza, *A coefficient of variation based image selective segmentation model using active contours*, East Asian J. Appl. Math. 2 (2012), pp. 150–169.

[7] R.E. Bank and D.J. Rose, *Analysis of a multilevel iterative method for nonlinear finite element equations*, Math. Comput. 39(160) (1982), pp. 453–465.

[8] K.J. Brabazon, M.E. Hubbard, and P.K. Jimack, *Nonlinear multigrid methods for second order differential operators with nonlinear diffusion coefficient*, Comput. Math. Appl. 68(12, Part A) (2014), pp. 1619–1634.

[9] D. Braess and W. Hackbusch, *A new convergence proof for the multigrid method including the v-cycle*, SIAM J. Numer. Anal. 20(5) (1983), pp. 967–975.

[10] J.H. Bramble, J.E. Pasciak, J.P. Wang, and J. Xu, *Convergence estimates for multigrid algorithms without regularity assumptions*, Math. Comput. 57(195) (1991), pp. 23–45.

[11] J.H. Bramble, J.E. Pasciak, J.P. Wang, and J. Xu, *Convergence estimates for product iterative methods with applications to domain decomposition*, Math. Comput.57(195) (1991), pp. 1–21.

[12] A. Brandt, *Multi-level adaptive solutions to boundary-value problems*, Math. Comput. 31(138) (1977), pp. 333–390.

[13] A. Brandt, Multi-level adaptive techniques for singular-perturbation problems. *NASA-TM-80966, ICASE report 78-18*, 1978.

[14] A. Brandt, *Rigorous quantitative analysis of multigrid, I. Constant coefficients two-level cycle with ℓ_2-norm*, SIAM J. Numer. Anal. 31(6) (1994), pp. 1695–1730.

[15] X. Bresson, S. Esedoglu, P. Vandergheynst, J.-P. Thiran, and S. Osher, *Fast global minimization of the active contour/snake model*, J. Math. Imaging Vis.28(2) (2007), pp. 151–167.

[16] W.L. Briggs, V. E. Henson, S.F. McCormick, *A Multigrid Tutorial* Vol. 72, SIAM, Philadelphia, USA, 2000.

[17] C. Brito-Loeza and K. Chen, *Multigrid algorithm for high order denoising*, SIAM J. Imaging Sci. 3(3) (2010), pp. 363–389.

[18] E.S. Brown, T.F. Chan, X. Bresson, *A convex relaxation method for a class of vector-valued minimization problems with applications to mumford-shah segmentation*. UcCLA Cam Rep., USA 10(43) (2010), pp. 1–34. ftp://ftp.math.ucla.edu/pub/camreport/cam10-43.pdf.

[19] V. Caselles, R. Kimmel, and G. Sapiro, *Geodesic active contours*, Int. J. Comput. Vis. 22(1) (1997), pp. 61–79.

[20] F. Catté, P.-L. Lions, J.-M. Morel, and T. Coll, *Image selective smoothing and edge detection by nonlinear diffusion*, SIAM J. Numer. Anal. 29(1) (1992), pp. 182–193.

[21] T.F. Chan and K. Chen, *On a nonlinear multigrid algorithm with primal relaxation for the image total variation minimisation*, Numer. Algorithms 41(4) (2006), pp. 387–411.

[22] T.F. Chan and L.A. Vese, *Active contours without edges*, IEEE Trans. Image Process. 10(2) (2001), pp. 266–277.

[23] T.F. Chan, S. Esedoglu, and M. Nikolova, *Algorithms for finding global minimizers of image segmentation and denoising models*, SIAM J. Appl. Math. 66(5) (2006), pp. 1632–1648.

[24] K. Chen, *Matrix Preconditioning Techniques and Applications*, Cambridge University Press, Cambridge, UK, 2005.

[25] D. Cremers, F.R. Schmidt, F. Barthel, Shape priors in variational image segmentation: Convexity, Lipschitz continuity and globally optimal solutions. *IEEE conference on Computer Vision and Pattern Recognition. CVPR 2008*, IEEE, Anchorage, Alaska, USA, 2008, pp. 1–6.

[26] P. D'Ambra and G. Tartaglione, *Solution of ambrosio-tortorelli model for image segmentation by generalized relaxation method*, Commun. Nonlinear Sci. Numer. Simul. 20 (2014), pp. 819–831.

[27] C.K. Filelis-Papadopoulos and G.A. Gravvanis, *On the multigrid method based on finite difference approximate inverses*, Comput. Model Eng. Sci. 90 (2013), pp. 233–253.

[28] C. Frohn-Schauf, S. Henn, and K. Witsch, *Nonlinear multigrid methods for total variation image denoising*, Comput. Vis. Sci. 7(3–4) (2004), pp. 199–206.

[29] B. Goldluecke, D. Cremers, Convex relaxation for multilabel problems with product label spaces. *European Conference on Computer Vision*, Springer, Crete, Greece, 2010, pp. 225–238.

[30] T. Goldstein and S. Osher, *The split bregman method for l1-regularized problems*, SIAM J. Imaging Sci. 2(2) (2009), pp. 323–343.

[31] D.G. Gordeziani and G.V. Meladze, *Simulation of the third boundary value problem for multidimensional parabolic equations in an arbitrary domain by one-dimensional equations*, USSR Comput. Math. Math. Phys. 14 (1974), pp. 249–253.

[32] C. Gout, C. Le Guyader, and L. Vese, *Image segmentation under interpolation conditions*. Preprint CAM-IPAM, 2003, p. 44.

[33] G.A. Gravvanis and C.K. Filelis-Papadopoulos, *On the multigrid cycle strategy with approximate inverse smoothing*, Eng. Comput. 31 (2014), pp. 110–122.

[34] W. Hackbusch, *Multi-grid Methods and Applications*, Springer Science & Business Media, Berlin Heidelberg, Germany, 2013.

[35] W. Hackbusch and A. Reusken, *Analysis of a damped nonlinear multilevel method*, Numer. Math.55(2) (1989), pp. 225–246.

[36] W. Hackbusch, A. Reusken, On global multigrid convergence for nonlinear problems. in *Robust Multi-grid Methods*, Proceedings of 4th GAMM-Seminar, Kiel 1988, W. Hackbusch, ed., Springer Vieweg Teubner Verlag, 1989, pp. 105–113.
[37] V.E. Henson, *Multigrid methods nonlinear problems: An overview*, Comput. Imaging 5016 (2003), pp. 36–48.
[38] P. Jaccard, *The distribution of the flora in the alpine zone. 1*, New Phytologist 11(2) (1912), pp. 37–50.
[39] A.K. Jumaat and K. Chen, *An optimization-based multilevel algorithm for variational image segmentation models*, Electron. Trans. Numer. Anal. 46 (2017), pp. 474–504.
[40] T. Kim, J.E. Pasciak, and P.S. Vassilevski, *Mesh-independent convergence of the modified inexact newton method for a second order non-linear problem*, Numer. Linear Algebra Appl. 13(1) (2006), pp. 23–47.
[41] M. Klodt, F. Steinbrücker, D. Cremers, *Moment constraints in convex optimization for segmentation and tracking*. Adv. Topics Comput. Vis., Advances in Computer Vision and Pattern Recognition (2013), pp. 1–29.
[42] C. Le Guyader and C. Gout, *Geodesic active contour under geometrical conditions: Theory and 3D applications*, Numer. Algorithms 48(1–3) (2008), pp. 105–133.
[43] R.J. Leveque and Z. Li, *The immersed interface method for elliptic equations with discontinuous coefficients and singular sources*, SIAM J. Numer. Anal. 31(4) (1994), pp. 1019–1044.
[44] C. Liu, M.K.-P. Ng, T. Zeng, *Weighted variational model for selective image segmentation with application to medical images*. Pattern Recognit. 76 (2018), pp. 367–379.
[45] T. Lu, P. Neittaanmäki, and X-C. Tai, *A parallel splitting up method and its application to navier-stokes equations*, Appl. Math. Lett. 4(2) (1991), pp. 25–29.
[46] S.P. MacLachlan and C.W. Oosterlee, *Local fourier analysis for multigrid with overlapping smoothers applied to systems of pdes*, Numer. Linear Algebra Appl. 18(4) (2011), pp. 751–774.
[47] A. Napov and Y. Notay, *Smoothing factor, order of prolongation and actual multigrid convergence*, Numer. Math. 118(3) (2011), pp. 457–483.
[48] T.N.A. Nguyen, J. Cai, J. Zhang, and J. Zheng, *Robust interactive image segmentation using convex active contours*, IEEE Trans. Image Process 21(8) (2012), pp. 3734–43.
[49] P. Ochs, T. Brox, Object segmentation in video: A hierarchical variational approach for turning point trajectories into dense regions. *International Conference on Computer Vision*, Barcelona, 2011, pp. 1583–1590. doi:10.1109/ICCV.2011.6126418.
[50] S. Osher and J. Sethian, *Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations*, J. Comput. Phys. 79(1) (1988), pp. 12–49.
[51] P. Perona and J. Malik, *Scale-space and edge detection using anisotropic diffusion*, IEEE Trans. Pattern Anal. Mach. Intell. 12(7) (1990), pp. 629–639.
[52] L. Rada and K. Chen, *A new variational model with dual level set functions for selective segmentation*, Commun. Comput. Phys. 12(1) (2012), pp. 261–283.
[53] L. Rada and K. Chen, *Improved selective segmentation model using one level-set*, J. Algorithms Comput. Technol. 7(4) (2013), pp. 509–540.
[54] A. Reusken, *Convergence of the multigrid full approximation scheme for a class of elliptic mildly nonlinear boundary value problems*, Numer. Math. 52(3) (1987), pp. 251–277.
[55] A. Reusken, *Convergence of the multilevel full approximation scheme including thev-cycle*, Numer. Math. 53(6) (1988), pp. 663–686.
[56] M. Roberts, *Selective image segmentation models and fast multigrid methods*. PhD thesis, University of Liverpool, 2019.
[57] M. Roberts and J. Spencer, *Chan-Vese reformulation for selective image segmentation*, J. Math. Imaging Vis. (2019), To appear.
[58] M. Roberts, K. Chen, K.L. Irion, *Multigrid algorithm based on hybrid smoothers for variational and selective segmentation models*. Int. J. Comput. Math. 96(8) (2019), pp. 1623–1647. doi:10.1080/00207160.2018.1494827.
[59] M. Roberts, K. Chen, and K.L. Irion, *A convex geodesic selective model for image segmentation*, J. Math. Imaging Vis. 61(4) (2019), pp. 482–503.
[60] V.V. Shaidurov, *Multigrid Methods for Finite Elements*, Springer Science & Business Media, Dordrecht, Netherlands, 2013.
[61] J. Spencer and K. Chen, *A convex and selective variational model for image segmentation*, Commun. Math. Sci. 13(6) (2015), pp. 1453–1472.
[62] C. Tai, X. Zhang, and Z. Shen, *Wavelet frame based multiphase image segmentation*, SIAM. J. Imaging Sci. 6(4) (2013), pp. 2521–2546.
[63] U. Trottenberg, C.W. Oosterlee, A. Schüller, *Multigrid*, Academic Press, London, UK, 2001.
[64] L.A. Vese and T.F. Chan, *A multiphase level set framework for image segmentation using the MS model*, Int. J. Comput. Vis. 50(3) (2002), pp. 271–293.
[65] J. Weickert, B.M. Ter Haar Romeny, and M.A. Viergever, *Efficient and reliable schemes for nonlinear diffusion filtering*, IEEE Trans. Image Process. 7(3) (1998), pp. 398–410.
[66] R. Wienands, W. Joppich, *Practical Fourier Analysis for Multigrid Methods*, CRC press, Boca Raton, Florida, USA, 2004.

[67] D. Xie, New nonlinear multigrid analysis. *NASA Conference Publication 3339*, Proceedings of Copper Mountain Conference on Multigrid Methods, N. D. Melson, T. A. Manteuffel, S. F. McCormick and C. C. Douglas, eds., Copper Mountain, Colorado, USA, 1996, pp. 793–808. http://citeseerx.ist.psu.edu/viewdoc/download?doi = 10.1.1.38.4442&rep = rep1&type = pdf.
[68] J. Xu, *Iterative methods by space decomposition and subspace correction*, SIAM Rev. 34(4) (1992), pp. 581–613.
[69] J. Zhang, K. Chen, and D.A. Gould, *A fast algorithm for automatic segmentation and extraction of a single object by active surfaces*, Int. J. Comput. Math.92(6) (2015), pp. 1251–1274.
[70] W. Zhu, X.-C. Tai, and T.F. Chan, *Image segmentation using euler's elastica as the regularization*, J. Sci. Comput. 57 (2013), pp. 414–438.