On a Nonlinear Multigrid Algorithm with Primal Relaxation for the Image Total Variation Minimisation

Tony F. Chan[*] and **Ke Chen**[†]

Abstract

Digital image restoration has drawn much attention in the recent years and a lot of research has been done on effective variational partial differential equation models and their theoretical studies. However there remains an urgent need to develop fast and robust iterative solvers, as the underlying problem sizes are large. This paper proposes a fast multigrid method using primal relaxations. The basic primal relaxation is known to get stuck at a 'local' non-stationary minimum of the solution, which is usually believed to be 'non-smooth'. Our idea is to utilize coarse level corrections, overcoming the deadlock of a basic primal relaxation scheme. A further refinement is to allow non-regular coarse levels to correct the solution, which helps to improve the multilevel method. Numerical experiments on both 1D and 2D images are presented.

AMS subject class: 68U10, 65F10, 65K10.

Keywords: Image restoration, Total variation, Regularisation, Primal relaxation, Nonlinear solvers.

Contents

[*] Department of Mathematics, University of California, Los Angeles, CA 90095-1555, USA. Email: chan@math.ucla.edu. Web: http://www.math.ucla.edu/∼chan

[†] Department of Mathematical Sciences, University of Liverpool, Peach Street, Liverpool L69 7ZL, UK. Email: k.chen@liverpool.ac.uk. Web: http://www.liv.ac.uk/∼cmchenke [for correspondence]

1 Introduction

Image restoration is usually modelled by a minimisation formulation in a variational setting [2, 4, 15, 36, 1, 42]. The ill-posed nature of the inverse problem is dealt with by regularization techniques. The most effective regularization technique is the so called total variation (TV) minimisation due to [36]. The resulting nonlinear partial differential equation (PDE), via the Euler-Lagrange solution, has an interesting coefficient (not defined near flat regions of the solution) which leads to convergence problems of many iterative solvers. This paper is concerned with the basic TV model [36]; other models are reviewed in the next section.

As far as solving the nonlinear PDE is concerned, the existing work on the topic falls into three categories: (i). Fixed point iteration [1, 40, 43, 44, 41, 42, 35]. Once the coefficients are fixed, various iterative solver techniques including multigrid methods have been considered [43, 44, 12, 13, 11, 29, 6, 20]. The robustness of a linear multigrid can be enhanced by algebraic multigrid methods (AMGs) [16, 5] but AMGs are known to be expensive. Further improvements are still useful. (ii). Explicit time marching scheme [36, 31] that turns the nonlinear PDE into a parabolic equation before using an explicit Euler method to march in time to (slow) convergence. (iii). Primal-dual method [15, 14, 4] that solves for both the primal and dual variables together in order to achieve faster convergence with the Newton method (and a constrained optimisation with the dual variable). One observes that all these methods require a small positive parameter β to avoid singularities in the PDE coefficient associated with flat regions of the solution and consequently are sensitive to the parameter. In a recent work, recognizing that primal relaxation alone does not work, Carter [7] tried the dual formulation that does not involve this parameter and achieved some initial success with the so-called barrier (interior point) dual relaxation method. Nonlinear multilevel methods for such problems only exist for a smoothed version of the underlying PDE [21, 38] or a smoothed version of the underlying merit functional [30].

This paper proposes an alternative method to solve *directly* the image minimisation problem to the PDE approach. The problems to overcome include the treatment of the non-differentiable functional in the minimisation problem, by using local minimisation (and hence dimension reduction), and the stagnation problem with the primal relaxation, by using coarse levels in a multilevel scheme. In Section 2, we introduce the image problem and the associated nonlinear equations before discussing some solution methods. In Section 3, we present the primal relaxation method and the algorithmic development details in a multilevel setting. In Section 4, we present some supporting numerical results for the denoising case.

2 Total variation regularization and solution

Denote by $u = u(x, y)$ the true image and $z = z(x, y)$ the observed image, both defined in the bounded square domain $\Omega = [0, 1] \times [0, 1] \subset \mathbf{R}^2$. In practice, only z is only available in a discrete (matrix) form [3, 19, 37]. The observed image z has been contaminated in data collection stage. The purpose of image restoration is to recover u as much as we can using a degradation model

$$u - z = \eta, \tag{1}$$

where η is a Gaussian white noise (unknown).

Image restoration is thus an inverse problem that may not have a unique solution. Some regularity condition has to be imposed on the solution space in order to turn the underlying problem to a well posed one [42]. We shall use the well known TV regularization to ensure that sharp features of an image are preserved [36]. However we note that there are many other regularization functionals (beyond the variational framework) that might be used; see [2, 28] and the references therein.

Below we briefly review the common methodology to set out the context of our algorithm in next sections. Following early work [15], we choose the Tikhonov direct regularization technique to solve the inverse problem (1)

$$\min_u J(u), \qquad J(u) = \overline{\alpha} R(u) + \frac{1}{2}\|u - z\|_2^2, \tag{2}$$

where the regularization functional $R(u)$ is selected as the TV-norm [36, 15]

$$R(u) = \int_\Omega |\nabla u| dxdy = \int_\Omega \sqrt{u_x^2 + u_y^2} dxdy. \tag{3}$$

Here the parameter $\overline{\alpha}$ represents a tradeoff between the quality of the solution and the fit to the observed data. Thus the overall image restoration problem is the following

$$\min_u J(u), \qquad J(u) = \int_\Omega \Big[\overline{\alpha}\sqrt{u_x^2 + u_y^2} + \frac{1}{2}(u - z)^2\Big] dxdy \tag{4}$$

or

$$\min_u \overline{\alpha}\Big\||\nabla u|\Big\|_{L_1} + \frac{1}{2}\Big\|u - z\Big\|_{L_2}^2. \tag{5}$$

The theoretical solution to problem (4) is given by the Euler-Lagrange equation (assuming homogeneous Neumann boundary conditions and $|\nabla u| \neq 0$)

$$\overline{\alpha}\nabla \cdot \Big(\frac{\nabla u}{|\nabla u|}\Big) - u = -z. \tag{6}$$

For 1D problems $\Omega = [0, 1]$ with $|\frac{du}{dx}| \neq 0$, the above equation reduces to

$$\overline{\alpha}\frac{d}{dx}\Big(\Big|\frac{du}{dx}\Big|^{-1}\frac{du}{dx}\Big) - u = -z. \tag{7}$$

Notice that the nonlinear coefficient may have a zero denominator in general i.e. the equation is not defined at points where $|\nabla u| = 0$ (corresponding to flat regions of the solution).

A commonly adopted idea to overcome this apparent difficulty was to introduce (yet) another parameter β to (4) and (6) so the new Euler-Lagrange equation (regardless $|\nabla u| = 0$ or not) becomes

$$\overline{\alpha}\nabla \cdot \Big(\frac{\nabla u}{\sqrt{|\nabla u|^2 + \beta}}\Big) - u = -z, \tag{8}$$

where corresponds to minimising, instead of (4),

$$\min_u J_\beta(u), \qquad J_\beta(u) = \int_\Omega \Big[\overline{\alpha}\sqrt{u_x^2 + u_y^2 + \beta} + \frac{1}{2}(u - z)^2\Big] dxdy \tag{9}$$

and in theory $u = u_\beta(x, y)$ differs from u in (6). Observe that when $\beta = 0$ equation (8) reduces to (6); moreover as $\beta \to 0$, $u_\beta \to u$ as shown in [1].

As mentioned in the introduction, there are three main solution methods for solving equation (8) — differing in how to deal with nonlinearities

- **Fixed point iteration** [1, 40, 43, 44, 41, 42]:
  Solve a lagged diffusion problem until $u^{k+1} - u^k$ is small

  $$\overline{\alpha}\nabla \cdot \Big(\frac{\nabla u^{k+1}}{\sqrt{|\nabla u^k|^2 + \beta}}\Big) - u^{k+1} = -z. \tag{10}$$

  There exist a large literature on this topic, mainly due to wide interest in developing fast iterative solvers for the above linear equations (once discretized). Further improvements on robustness of these solvers are still needed.

- **Explicit time marching scheme** [36, 31]:
  The original idea in [36] was refined in [31] as solving the following parabolic PDE until a steady state has been reached
  $$u_t = |\nabla u|\Big[\overline{\alpha}\nabla \cdot \Big(\frac{\nabla u}{\sqrt{|\nabla u|^2+\beta}}\Big) - u + z\Big]. \tag{11}$$
  As remarked in [33], for linear problems, this type of ideas represents a kind of relaxation schemes. The drawback may be the artificial time step Δt must be small due to stability requirement.

- **Primal-dual method** [15, 14, 4]:
  As discussed in [4], Newton method for equation (8) leads to very slow or no convergence because z is often not a sufficiently close initial guess for u. Introducing the dual variable (vector) $\omega = \nabla u/\sqrt{|\nabla u|^2+\beta}$ appears to have made the combined system
  $$\begin{cases} \overline{\alpha}\nabla \cdot \omega - u = -z, \\ \omega\sqrt{|\nabla u|^2+\beta} - \nabla u = 0. \end{cases}$$
  in two variables (u, ω) more amenable to Newton iterations as the new system is nearly "linear" in the two variables (as observed in [15]). Note that ω is constrained in each iteration step so the overall algorithm needs some care in any implementation.

Remark 1 *If one is contented with the system (8) with parameter β, the recommended approach is the CGM although there is scope to realize the implementation more efficiently. A simple observation of equation (6) reveals that there is nothing wrong with it in the sense that the equation is still "non-singular" (though towards undefined) when nonzero $|\nabla u|$ gets small as $\omega = \nabla u/|\nabla u|$ is always bounded. This suggests that it may not be an appropriate action to take in introducing the parameter β dependent on the size of $|\nabla u|$ which is a free gradient otherwise. For fixed point iterations, this point is more pronounced since the 'linearized' PDE (10) has large coefficients near points where the solution is flat (or less interesting!), creating somewhat "unnecessary" jumps in coefficients. See [26, 34] for theoretical arguments on problems with including β.*

Remark 2 *Image denoising is related to the deblurring problem: $z = Ku + \eta$ modelled by*
$$\min_u J(u), \qquad J(u) = \overline{\alpha}R(u) + \frac{1}{2}\|Ku - z\|_2^2, \tag{12}$$
(compare to (2)) where K is a blurring (integral) operator. When K is a convolution operator, the challenge is to solve resulting linear system without forming the discretized matrix of K^*K (mimicking the capability of the fast multipole method) [43, 44, 12, 13, 11, 29]. Generalization of our multigrid work to the deblurring problem and other imaging problems is currently in progress.

Before we proceed, we remark on related work towards efficient solution of the denoising model (2) without introducing β. There exist at least six approaches worth pursuing:

- **The dual formulation of (2).** The primal variable u as solution of (2) can be replaced [9, 22] by its dual variable $\mathbf{p}$ which is sought in a new optimization problem with a differentiable merit function; here $u = z - \overline{\alpha}\nabla\mathbf{p}$. Further time-marching schemes may be used efficiently [9].

- **The dual formulation of an approximate form of (2).** When the TV norm is approximated, a Fenchel dual formulation [23] can be derived and further non-smooth Newton methods are applicable.

- **The combined method with a smooth functional to compensate for 'flat' regions**. If $|\nabla u| \neq 0$ in (2), the model is easy to solve. For the general case, one can modify [10, 27, 34] the TV-norm to exclude all sets where $|\nabla u| = 0$. As compensation, regularization over these sets is done with smooth norms such as with $|\nabla u|^2$.
- **The active set method**. This is a discrete approach [25, 8, 26] solving the Euler-Lagrange equation which implements a related idea to the above combined method i.e. treat active sets ($|\nabla u| = 0$) differently from inactive sets.
- **The tube method**. The discrete solution of (2) lies in a tube-like domain. The work of [24] based on the Taut-string algorithm proposes a constrained optimization method as another primal-dual approach. Different from [15], no β is required. However a fixed-point algorithm (outer-loop) is required in solving the nonlinear optimization.
- **The second-order cone programming method**. This is a direct method to solve (2) numerically. As shown in [18], the idea is to convert the discretized version of (2) to a constrained optimization problem which is in turn solved by the interior point algorithm that allows non-differentiable cone constraints (with the TV terms turned into cone conditions). Here we have 3 times more unknowns and the overall complexity is $O(N\sqrt{N})$ with $N = n^2$ for an $n \times n$ image.

Although ideas from this paper may be generalized to the above new or reformulated models in a multilevel setting, such generalization work remains to be done.

In what follows we shall solve the original primal and optimization formulation (4) which has no additional parameters such as β, by a multilevel method. As a multilevel method is more optimal than unilevel methods, its complexity is only $O(N \log N)$ with $N = n^2$ for an $n \times n$ image.

3 Piecewise constant primal relaxation and multilevel schemes

Instead of pursuing fast solvers for solving (6) (involving β) as previously done in [11, 40, 43, 41, 42, 16, 38], we consider how to design fast algorithms for solving the minimisation problem (4) without β. To this end, consider the discretized form of (4) respectively in 1D and 2D:

$$\min_{u \in \mathbf{R}^n} J(u) = \min_{u \in \mathbf{R}^n} \alpha \sum_{j=1}^{n-1} |u_j - u_{j+1}| + \frac{1}{2} \sum_{j=1}^{n} (u_j - z_j)^2 \tag{13}$$

and

$$\min_{u \in \mathbf{R}^{n \times n}} J(u) = \min_{u \in \mathbf{R}^n} \alpha \sum_{i=1}^{n-1} \sum_{j=1}^{n-1} \sqrt{(u_{ij} - u_{i,j+1})^2 + (u_{ij} - u_{i+1,j})^2} + \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} (u_{ij} - z_{ij})^2, \tag{14}$$

with $\alpha = \overline{\alpha}/h$ and $h = 1/(n-1)$.

Our objective is to solve (4) via (13) and (14), not (9), using a multilevel method. Our result as intended shows that the solution in some cases is sharper than that from (9) for problems where features such as edges and corners are important to preserve. Below we shall first expose what the known problems with optimization methods are before proposing our computational algorithms.

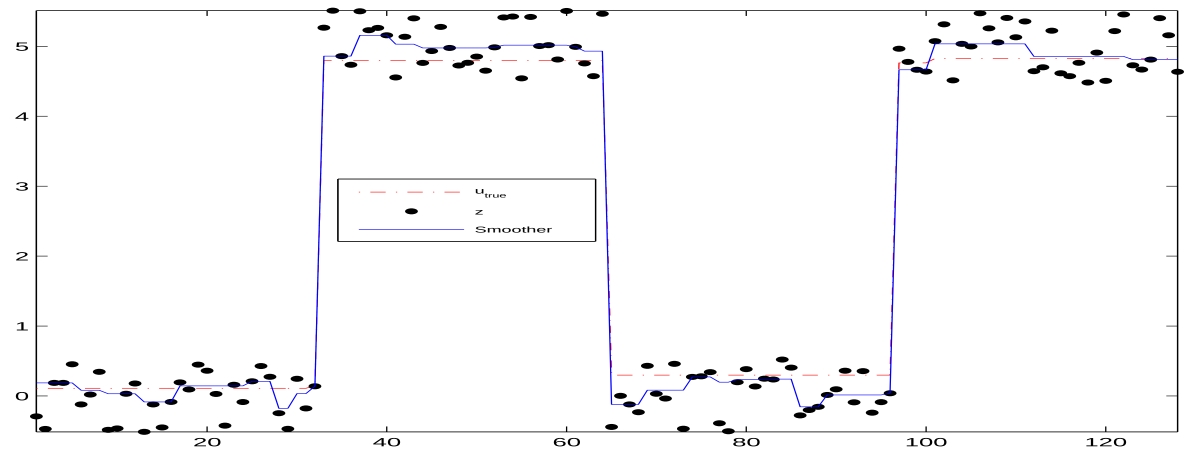
3.1 Smoothing and local minimisations

Consider first the solution of (13) ie. (4) in the 1D case. An important issue to address is how to make use of an approximate solution $\widetilde{u}$ for further solution or how to measure the "residual" for improved smoothing (or relaxation) in the minimisation problem . That is to say, here, there is no obvious way to define a residual correction functional (which is easily done for an operator equation [17]). A local minimisation problem at a typical node j is the following

$$\min_{u_j \in \mathbf{R}} J_j(u_j) = \min_{u_j} \alpha|\widetilde{u}_{j-1} - u_j| + \alpha|u_j - \widetilde{u}_{j+1}| + \frac{1}{2}(u_j - z_j)^2 \tag{15}$$

which can be solved analytically because the local solution u_j is taken from the set $\{z_j - 2\alpha,\ z_j + 2\alpha,\ \widetilde{u}_{j-1},\ \widetilde{u}_{j+1},\ z_j\}$. The 2D case is similar but the solution involves iterations (as shown later). If we solve for all u_j, $j = 1, 2, \ldots, n$, in (15) for several steps, then the resulting solution u is smooth in some sense but, unfortunately, the method of smoothing will converge (quickly) to the wrong solution as the number of steps increases. This is known as the 'local minimiser' getting 'stuck'; see [7]. Here by a '*local minimiser*' (in quotes), we mean the minimizer obtained from local coordinate direction minimisations which is different from the local minimiser (which is also the global minimizer). Figure 1 shows a typical example where z_j's are shown as dots ('$\cdot$'), the wrong solution u is shown as the solid line while the dash-dotted line denotes the true global minimiser. Clearly the 'local' minimisers get stuck.

Figure 1: An example showing that local minimisations get stuck at the incorrect solution (solid line). Here $n = 128$, $\alpha = 4$, z is shown as $\bullet$ and the dash-dotted line is the true solution.



To see the problem more clearly, we use a small datum set with $n = 8$ and show the result in Figure 2. There one observes that the incorrect solution ('∇') only has the wrong 'heights', when compared to the true solution ('$\Box$'). This motivates us the following multilevel method to correct such 'wrong' heights.

3.2 A piecewise constant based multilevel method

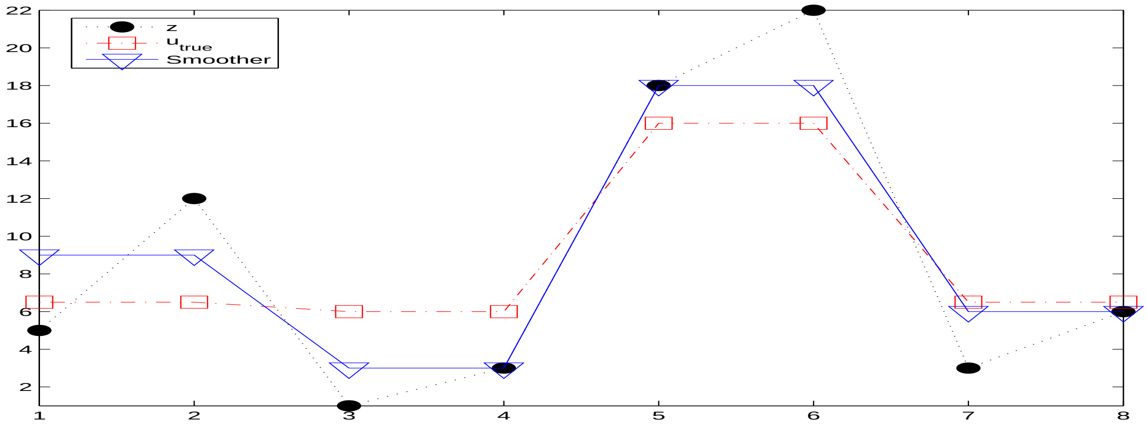
Assume that $\widetilde{u} \in \mathbf{R}^n$ is our current approximation to (4). We wish to find the best piecewise constant function $c \in \mathbf{R}^n$ so that the following holds

$$\min_c J(\widetilde{u} + c). \tag{16}$$

Such a problem is as 'difficult' as the original problem (4). Therefore we solve the correction problem

$$\min_c J(\widetilde{u} + c), \qquad c = P_k c^{(k)} \in \mathbf{R}^n, \quad c^{(k)} \in \mathbf{R}^{\tau_k} \tag{17}$$

Figure 2: A second example showing that local minimisations get stuck at the incorrect solution ($\bigtriangledown$). Here $n = 8$, $\alpha = 4$, z is shown as $\bullet$ and the true solution as $\Box$.



where $\tau_k = n/2^{k-1}$ for $k = 1, 2, \ldots, L+1$ and $P_k : \mathbf{R}^{\tau_k} \to \mathbf{R}^n$ is the interpolation operator for piecewise constants. Normally we may take $n = 2^L$ so $L = \log_2 n$ and the coarsest level has $\tau_{L+1} = 1$, corresponding to a single constant. Thus on the finest level $\tau_1 = n$, the smoothed solution of (17) can be obtained as in the previous subsection.

We now formulate the above problem (17) on a general level k so that the previous smoothing method can be applied. Let

$$c^{(k)} = [c_1 \; c_2 \; \cdots \; c_{\tau_k}]^T.$$

Then

$$c = P_k c^{(k)} = [\underbrace{c_1 \;\cdots c_1}_{\text{block } 1} \; \underbrace{c_2 \;\cdots c_2}_{\text{block } 2} \; \cdots \; \cdots \; \underbrace{c_{\tau_k} \;\cdots c_{\tau_k}}_{\text{block } \tau_k}]^T \tag{18}$$

and, on substituting into (17) and defining $b = n/\tau_k = 2^{k-1}$, we can reformulate the functional as the following

$$\begin{aligned}
& J(\widetilde{u} + c) \\
= \;& \alpha \Big[|\widetilde{u}_1 + c_1 - (\widetilde{u}_2 + c_1)| + \cdots + |\widetilde{u}_b + c_1 - (\widetilde{u}_{b+1} + c_2)| + \\
& \quad |\widetilde{u}_{b+1} + c_2 - (\widetilde{u}_{b+2} + c_2)| \; + \cdots + \; |\widetilde{u}_{2b} + c_2 - (\widetilde{u}_{2b+1} + c_3)| \; + \\
& \qquad \cdots \; + \\
& \quad |\widetilde{u}_{n-b+1} + c_{\tau_k} - (\widetilde{u}_{n-b+2} + c_{\tau_k})| + \cdots + |\widetilde{u}_{n-1} + c_{\tau_k} - (\widetilde{u}_n + c_{\tau_k})| \Big] \\
& + \frac{1}{2} \sum_{\ell=1}^{\tau_k} \sum_{j=(\ell-1)b+1}^{\ell b} (\widetilde{u}_j + c_j - z_j)^2 \\
= \;& \alpha \sum_{j=1}^{\tau_k - 1} \Big| \widetilde{u}_{jb} - \widetilde{u}_{jb+1} + c_j - c_{j+1} \Big| + \frac{1}{2} \sum_{\ell=1}^{\tau_k} \sum_{j=(\ell-1)b+1}^{\ell b} (\widetilde{u}_j + c_\ell - z_j)^2 + J_0(\widetilde{u}) \\
= \;& \alpha \sum_{j=1}^{\tau_k - 1} \Big| \widetilde{d}_{j+1} + c_j - c_{j+1} \Big| + \frac{1}{2} \sum_{\ell=1}^{\tau_k} \sum_{j=(\ell-1)b+1}^{\ell b} (c_\ell - \widetilde{z}_j)^2 + J_0(\widetilde{u})
\end{aligned}$$

$$= \alpha \sum_{j=1}^{\tau_k-1} \left|\overline{c}_j - \overline{c}_{j+1}\right| + \frac{b}{2}\sum_{\ell=1}^{\tau_k}(c_\ell - \widetilde{w}_\ell)^2 + J_0(\widetilde{u}) + J_1(\widetilde{u}) \tag{19}$$

$$= \alpha \sum_{j=1}^{\tau_k-1} \left|\overline{c}_j - \overline{c}_{j+1}\right| + \frac{b}{2}\sum_{j=1}^{\tau_k}(\overline{c}_j - \overline{z}_j)^2 + J_0(\widetilde{u}) + J_1(\widetilde{u})$$

$$= b\Big\{ \underbrace{\frac{\alpha}{b} \sum_{j=1}^{\tau_k-1} \left|\overline{c}_j - \overline{c}_{j+1}\right| + \frac{1}{2}\sum_{j=1}^{\tau_k}(\overline{c}_j - \overline{z}_j)^2}_{\text{Coarse level equation with a different "}\alpha\text{"}} + J_0(\widetilde{u})/b + J_1(\widetilde{u})/b\Big\} \tag{20}$$

where we have isolated the terms associated with c from the others not involving c. In particular, $J_0(\widetilde{u})$ and $J_1(\widetilde{u})$ denote those remaining terms not depending on c, $\widetilde{d}_{j+1} = \widetilde{u}_{jb} - \widetilde{u}_{jb+1}$ (with $\widetilde{d}_1 = 0$) denotes the jth jump between two adjacent blocks and, with

$$d_j = \sum_{\ell=1}^{j} \widetilde{d}_\ell, \qquad \text{for } j = 1, 2, \ldots, \tau_k,$$

the new quantities are (for $j = 1, 2, \ldots, \tau_k$)

$$\overline{c}_j = c_j - d_j, \qquad \overline{z}_j = \widetilde{w}_j - d_j. \tag{21}$$

Here with $\widetilde{z}_j = z_j - \widetilde{u}_j$ for $j = 1, 2, \ldots, n$, the mean 'z' value for each block is

$$\widetilde{w}_\ell = \sum_{j=(\ell-1)b+1}^{\ell b} \widetilde{z}_j / b \tag{22}$$

and the equation in (19) is obtained from using the simple equality of the type

$$\sum_{j=1}^{b}(v - z_j)^2 = bv^2 - 2(\sum_{j=1}^{b} z_j)v + \sum_{j=1}^{b} z_j^2 = b\Big(v - \overline{z}\Big)^2 + \sum_{j=1}^{b} z_j^2 - b\overline{z}^2,$$

where $\overline{z} = \sum_{j=1}^{b} z_j / b$.

Thus from (20), we see that minimizing the finest level correction problem associated with level k is equivalent to solving the following level k correction problem

$$\min_{\overline{c} \in \mathbf{R}^{\tau_k}} \frac{\alpha}{b} \sum_{j=1}^{\tau_k-1} \left|\overline{c}_j - \overline{c}_{j+1}\right| + \frac{1}{2}\sum_{j=1}^{\tau_k}(\overline{c}_j - \overline{z}_j)^2 \tag{23}$$

whose solution defines the correction vector $c = \overline{c} + d$ (note $b = 2^{k-1}$). As with most multilevel methods, we shall not solve (23) exactly. Instead we apply the smoothing method of the previous subsection §3.1 and then apply the idea repeatedly with coarser levels until the coarsest level.

We now summarize the overall multilevel algorithm as follows.

Algorithm 1 *Given z and an initial guess $\widetilde{u} = z$, with $L + 1$ levels,*

(1) Let $u_0 = \widetilde{u}$.

(2) Smooth the approximation on the finest level 1, i.e. solve (15) for $j = 1, 2, \ldots, n$.

(3) On coarse levels $k = 2, 3, \ldots, L + 1$:

— compute $\widetilde{z} = z - \widetilde{u}$ via (22)

— compute the local mean $\widetilde{w}_\ell = mean(\widetilde{z}\Big((\ell - 1)b + 1 : \ell b)\Big)$ via (22)

— compute $\widetilde{d}_j, d_j, \overline{c}_j, \overline{z}_j$ via (21)

— solve (23) by solving local minimisation as in (15) if $k \leq L$ or

— on the coarsest level $k = L + 1$, the correction constant is simply

$$\overline{c} = mean(\overline{z}) = c = mean(z - \widetilde{u}).$$

— Add the correction, $\widetilde{u} = \widetilde{u} + P_k c$ via (17).

(4) If $\|\widetilde{u} - u_0\|_2$ is small enough, stop or return to Step (1).

3.3 The variable patch coarse level correction

The above Algorithm 1 works well for many examples. However there exist difficult cases where the standard coarsening proves to be inadequate as illustrated by Figure 3, where for $n = 8$ and the initial $\widetilde{u}$ after relaxations on level 1, the next block size $b = 2 = 2^{2-1}$ on level $k = 2$ is not suitable (or rather a mixture of $b = 2, 3$ blocks is better).

Without using an algebraic multigrid method for 'automatic' coarsening (as in [16, 39]), we propose a flat patch based coarsening idea to supplement the use of a standard coarsening. Our idea is consisted of two parts. Firstly we divide the current approximation $\widetilde{u}$ into a union of m flat patches of respective lengths $b_1, b_2, \ldots, b_m$ (with $b_j \geq 1$). Secondly we take an extra coarse level with m piecewise constants, as before, in (17) and (18). Let

$$c^{<m>} = [c_1 \; c_2 \; \cdots \; c_m]^T$$

and

$$c = P_m c^{<m>} = [\underbrace{c_1 \; \cdots c_1}_{\text{block } b_1} \; \underbrace{c_2 \; \cdots c_2}_{\text{block } b_2} \; \cdots \; \cdots \; \underbrace{c_m \; \cdots c_m}_{\text{block } b_m}]^T \tag{24}$$

For solving $\min_c J(\widetilde{u} + c)$, one can derive an equation for c precisely as before with the only change of computing the local mean using b_j instead of b.

With this extra coarse level, Algorithm 1 can solve the problem in Figure 3 accurately as shown in Figure 4.

3.4 The 2D multilevel algorithm

The above ideas are applicable to higher dimensions. Most (single level) solution and algorithmic details for solving (14) or (4) in two dimensions can be found in Carter [7]:

$$\min_{u \in \mathbf{R}^n} J(u) = \min_u \alpha \sum_{i,j=1}^{n} \sqrt{(u_{ij} - u_{i+1,j})^2 + (u_{ij} - u_{i,j+1})^2} + \frac{1}{2} \sum_{i,j=1}^{n} (u_{ij} - z_{ij})^2, \tag{25}$$

where, due to Neumann's boundary condition for the continuous formulation, those *terms* involving indices larger than n (representing first derivatives in the first sum) are zero e.g. when $i = 2, j = n$,

$$\sqrt{(u_{2n} - u_{3,n})^2 + (u_{2n} - u_{2,n+1})^2} = |u_{2n} - u_{3,n}|.$$

Our emphasis below will be on a multilevel formulation.

Figure 3: An example showing that the standard coarsening can be inadequate. Here $n = 8$, $\alpha = 4$, z is shown as $\bullet$ and the true solution as $\Box$ while the standard coarsened multilevel solution $\triangle$ does not improve much from the initial 'stuck' solution $\triangledown$.

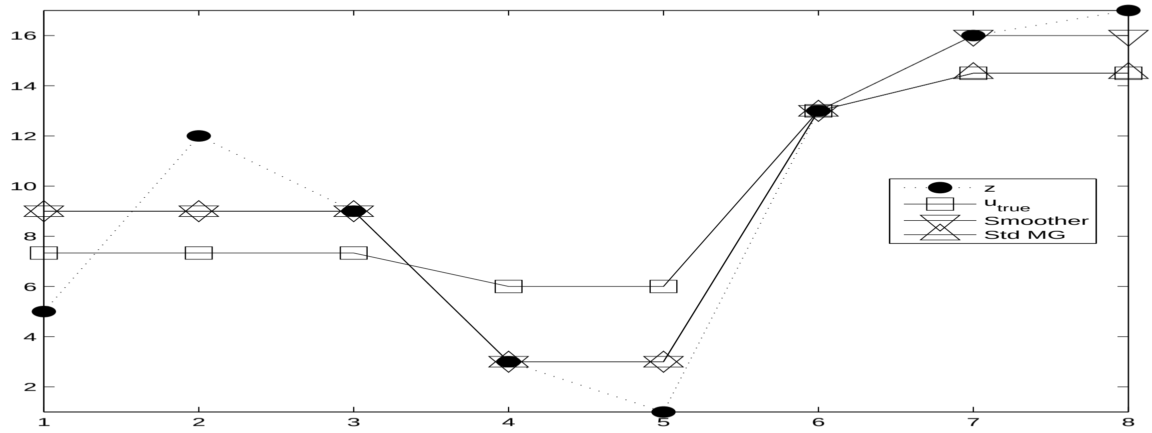
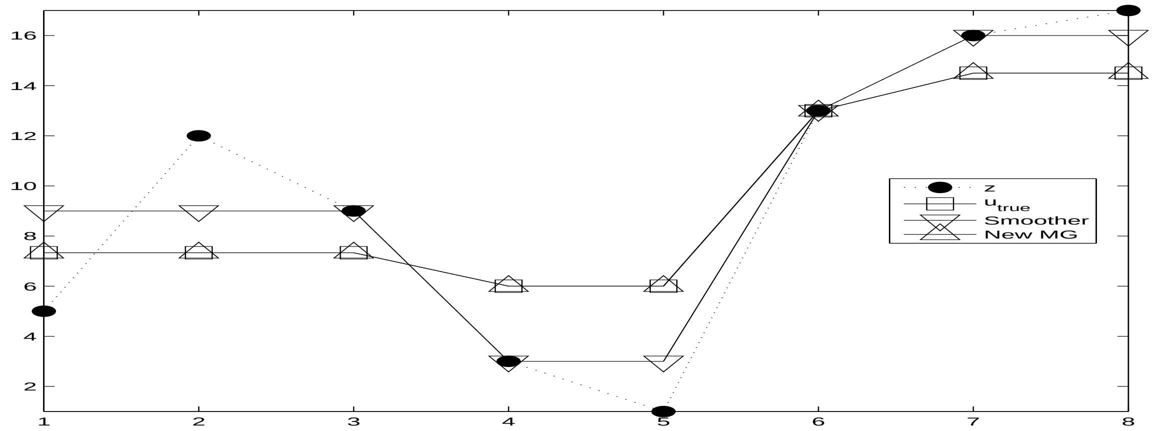


Figure 4: The example showing that the additional coarse level idea can add robustness to Alg. 1 (compare to Figure 3). Here $n = 8$, $\alpha = 4$, z is shown as $\bullet$ and the true solution as $\Box$ while the new multilevel solution $\triangle$ does improve on the initial 'stuck' solution $\triangledown$.



Point-wise Gauss-Seidel smoothing. Given an approximation $\widetilde{u}$, the smoothing idea of §3.1 can be used here to compute the minimizer for a local minimization problem [7]

$$\begin{aligned}\min_{u_{ij}} J_{ij}(u_{ij}) = \min_{u_{ij}} \alpha \Big[&\sqrt{(u_{ij}-\widetilde{u}_{i+1,j})^2+(u_{ij}-\widetilde{u}_{i,j+1})^2} + \\ &\sqrt{(u_{ij}-\widetilde{u}_{i-1,j})^2+(\widetilde{u}_{i-1,j}-\widetilde{u}_{i-1,j+1})^2} + \\ &\sqrt{(u_{ij}-\widetilde{u}_{i,j-1})^2+(\widetilde{u}_{i,j-1}-\widetilde{u}_{i+1,j-1})^2}\ \Big] + \\ &\frac{1}{2}(u_{ij}-z_{ij})^2.\end{aligned} \tag{26}$$

Different from the 1D case, the above problem does not have a convenient and analytical solution. For this type of equations, the Newton method is known to converge slowly [4] when α is relatively large.

We use the Richardson iteration to solve (26) from repeating $\widetilde{u} \to u \to \widetilde{u}$

$$\begin{aligned}\alpha\Big[&\frac{2u_{ij}-\widetilde{u}_{i+1,j}-\widetilde{u}_{i,j+1}}{\sqrt{(u_{ij}-\widetilde{u}_{i+1,j})^2+(u_{ij}-\widetilde{u}_{i,j+1})^2+\gamma}} + \\ &\frac{u_{ij}-\widetilde{u}_{i-1,j}}{\sqrt{(u_{ij}-\widetilde{u}_{i-1,j})^2+(\widetilde{u}_{i-1,j}-\widetilde{u}_{i-1,j+1})^2+\gamma}} + \\ &\frac{u_{ij}-\widetilde{u}_{i,j-1}}{\sqrt{(u_{ij}-\widetilde{u}_{i,j-1})^2+(\widetilde{u}_{i,j-1}-\widetilde{u}_{i+1,j-1})^2+\gamma}}\ \Big] + (u_{ij}-z_{ij}) = 0,\end{aligned}$$

where $\gamma > 0$ is a regularizing parameter which is much taken smaller than β above (e.g. $\gamma = 10^{-30}$ which is far too small for β in (8)). Then our Richardson iterations for solving (26) take the form

$$u_{ij}^{new} = R^{old} \Big/ L^{old} \tag{27}$$

where

$$\begin{aligned}L^{old} &= 2\alpha \Big/ \sqrt{(u_{ij}^{old}-\widetilde{u}_{i+1,j})^2+(u_{ij}^{old}-\widetilde{u}_{i,j+1})^2+\gamma} + \\ &+\alpha \Big/ \sqrt{(u_{ij}^{old}-\widetilde{u}_{i-1,j})^2+(\widetilde{u}_{i-1,j}-\widetilde{u}_{i-1,j+1})^2+\gamma} + \\ &+\alpha \Big/ \sqrt{(u_{ij}^{old}-\widetilde{u}_{i,j-1})^2+(\widetilde{u}_{i,j-1}-\widetilde{u}_{i+1,j-1})^2+\gamma}\ \Big] + 1, \\ R^{old} &= \alpha(\widetilde{u}_{i+1,j}+\widetilde{u}_{i,j+1}) \Big/ \sqrt{(u_{ij}^{old}-\widetilde{u}_{i+1,j})^2+(u_{ij}^{old}-\widetilde{u}_{i,j+1})^2+\gamma} + \\ &+\alpha\widetilde{u}_{i-1,j} \Big/ \sqrt{(u_{ij}^{old}-\widetilde{u}_{i-1,j})^2+(\widetilde{u}_{i-1,j}-\widetilde{u}_{i-1,j+1})^2+\gamma} + \\ &+\alpha\widetilde{u}_{i,j-1} \Big/ \sqrt{(u_{ij}^{old}-\widetilde{u}_{i,j-1})^2+(\widetilde{u}_{i,j-1}-\widetilde{u}_{i+1,j-1})^2+\gamma}\ \Big] + z_{ij}.\end{aligned}$$

An alternative to the Richardson iterations is the Newton method, which we found to perform similarly.

As an iterative method, the above iterations still converge to some non-stationary minimizer as in the 1D case (§3.1, as noted in [7]). And we shall use the multilevel method to rectify and improve the minimizer.

The local minimisation on a general level k. Similar to the 1D case, let $n = 2^L$ for $z \in \mathbf{R}^{n\times n}$ and the standard coarsening be used giving rise to $L+1$ levels $k = 1(finest), 2, \ldots, L, L+1(coarsest)$. Denote the dimension of level k by $\tau_k \times \tau_k$ with $\tau_k = n/2^{k-1}$. The generalization of Algorithm 1 to the 2D prompts us to consider

$$\min_{C=P_k\hat{c}\in\mathbf{R}^{n\times n},\hat{c}\in\mathbf{R}^{\tau_k\times\tau_k},c\in\mathbf{R}} J(\widetilde{u}+C), \tag{28}$$

where $P_k : \mathbf{R}^{\tau_k\times\tau_k} \to \mathbf{R}^{n\times n}$ is the interpolation operator. Let $c = (c_{ij})$. Then the computational stencil involving c_{ij} on level k can be shown as follows (set $b = 2^{k-1},\ k_1 = (i-1)b+1,\ k_2 = ib,\ \ell_1 =$

$(j-1)b+1,\ \ell_2 = jb)$

$$\begin{array}{c|ccc|c}
\vdots & \vdots & \cdots & \vdots & \vdots \\
\widetilde{u}_{k_1-1,\ell_2+1} + c_{i-1,j+1} & \widetilde{u}_{k_1,\ell_2+1} + c_{i,j+1} & \cdots & \widetilde{u}_{k_2,\ell_2+1} + c_{i,j+1} & \widetilde{u}_{k_2+1,\ell_2+1} + c_{i+1,j+1} \\
\hline
\widetilde{u}_{k_1-1,\ell_2} + c_{i-1,j} & \widetilde{u}_{k_1,\ell_2} + \mathbf{c}_{ij} & \cdots & \widetilde{u}_{k_2,\ell_2} + \mathbf{c}_{ij} & \widetilde{u}_{k_2+1,\ell_2} + c_{i+1,j} \\
\cdots & \vdots & \cdots & \vdots & \cdots \\
\widetilde{u}_{k_1-1,\ell_1} + c_{i-1,j} & \widetilde{u}_{k_1,\ell_1} + \mathbf{c}_{ij} & \cdots & \widetilde{u}_{k_2,\ell_1} + \mathbf{c}_{ij} & \widetilde{u}_{k_2+1,\ell_1} + c_{i+1,j} \\
\hline
\widetilde{u}_{k_1-1,\ell_1-1} + c_{i-1,j-1} & \widetilde{u}_{k_1,\ell_1-1} + c_{i,j-1} & \cdots & \widetilde{u}_{k_2,\ell_1-1} + c_{i,j-1} & \widetilde{u}_{k_2+1,\ell_1-1} + c_{i+1,j-1} \\
\vdots & \vdots & \cdots & \vdots & \vdots
\end{array} \tag{29}$$

Then, the local block (i,j) minimization for c_{ij} is equivalent to minimizing the following (resulting from the four sides of the above stencil - see (25))

$$\begin{aligned}
F(c_{ij}) = \alpha \sum_{\ell=\ell_1}^{\ell_2} & \sqrt{[c_{ij} - (\widetilde{u}_{k_1-1,\ell} - \widetilde{u}_{k_1,\ell})]^2 + (\widetilde{u}_{k_1-1,\ell} - \widetilde{u}_{k_1-1,\ell+1})^2} + \\
& \alpha \sum_{k=k_1}^{k_2-1} \sqrt{[c_{ij} - (\widetilde{u}_{k,\ell_2+1} - \widetilde{u}_{k,\ell_2})]^2 + (\widetilde{u}_{k,\ell_2} - \widetilde{u}_{k+1,\ell_2})^2} + \\
& \alpha \sqrt{[c_{ij} - (\widetilde{u}_{k_2,\ell_2+1} - \widetilde{u}_{k_2,\ell_2})]^2 + [c_{ij} - (\widetilde{u}_{k_2+1,\ell_2} - \widetilde{u}_{k_2,\ell_2})]^2} + \\
& \alpha \sum_{\ell=\ell_1}^{\ell_2-1} \sqrt{[c_{ij} - (\widetilde{u}_{k_2+1,\ell} - \widetilde{u}_{k_2,\ell})]^2 + (\widetilde{u}_{k_2,\ell} - \widetilde{u}_{k_2,\ell+1})^2} + \\
\alpha \sum_{k=k_1}^{k_2} \sqrt{[c_{ij} - (\widetilde{u}_{k,\ell_1-1} - \widetilde{u}_{k,\ell_1})]^2 + (\widetilde{u}_{k,\ell_1-1} - \widetilde{u}_{k+1,\ell_1-1})^2} & + \frac{1}{2} \sum_{k=k_1}^{k_2} \sum_{\ell=\ell_1}^{\ell_2} \Big[c_{k\ell} - (z_{k\ell} - \widetilde{u}_{k\ell}) \Big]^2.
\end{aligned} \tag{30}$$

To simplify the formulation, we define

$$\begin{aligned}
& \widetilde{z}_{k,\ell} = z_{k,\ell} - \widetilde{u}_{k,\ell}, \quad \widetilde{w}_{ij} = mean\Big(\widetilde{z}(k_1:k_2, \ell_1:\ell_2)\Big) = \frac{1}{b^2} \sum_{k=k_1}^{k_2} \sum_{\ell=\ell_1}^{\ell_2} \widetilde{z}(k,\ell), \\
& \widetilde{v}_{k,\ell} = \widetilde{u}_{k,\ell+1} - \widetilde{u}_{k,\ell}, \quad \widetilde{h}_{k,\ell} = \widetilde{u}_{k+1,\ell} - \widetilde{u}_{k,\ell}.
\end{aligned} \tag{31}$$

To also simplify the third term in (30), we may use the simple equality

$$\sqrt{(c-a)^2 + (c-b)^2} = \sqrt{2} \sqrt{\Big(c - \frac{a+b}{2}\Big)^2 + \Big(\frac{a-b}{2}\Big)^2}.$$

Therefore we can rewrite (30) as

$$\begin{aligned}
F(c_{ij}) &= \alpha \sum_{\ell=\ell_1}^{\ell_2} \sqrt{(c_{ij} - h_{k_1-1,\ell})^2 + v_{k_1-1,\ell}^2} \ + \ \alpha \sum_{k=k_1}^{k_2-1} \sqrt{(c_{ij} - v_{k,\ell_2})^2 + h_{k,\ell_2}^2} + \\
& \alpha \sum_{\ell=\ell_1}^{\ell_2-1} \sqrt{(c_{ij} - h_{k_2,\ell})^2 + v_{k_2,\ell}^2} + \alpha \sum_{k=k_1}^{k_2} \sqrt{(c_{ij} - v_{k,\ell_1-1})^2 + v_{k,\ell_1-1}^2} + \\
& \alpha \sqrt{2} \sqrt{(c_{ij} - \overline{v}_{k_2,\ell_2})^2 + \overline{h}_{k_2,\ell_2}^2} + \frac{1}{2} \sum_{k=k_1}^{k_2} \sum_{\ell=\ell_1}^{\ell_2} \Big(c_{k\ell} - \widetilde{z}_{k\ell} \Big)^2 \\
&= \alpha \sum_{\ell=\ell_1}^{\ell_2} \sqrt{(c_{ij} - h_{k_1-1,\ell})^2 + v_{k_1-1,\ell}^2} \ + \ \alpha \sum_{k=k_1}^{k_2-1} \sqrt{(c_{ij} - v_{k,\ell_2})^2 + h_{k,\ell_2}^2} + \\
& \alpha \sum_{\ell=\ell_1}^{\ell_2-1} \sqrt{(c_{ij} - h_{k_2,\ell})^2 + v_{k_2,\ell}^2} + \alpha \sum_{k=k_1}^{k_2} \sqrt{(c_{ij} - v_{k,\ell_1-1})^2 + v_{k,\ell_1-1}^2} + \\
& \alpha \sqrt{2} \sqrt{(c_{ij} - \overline{v}_{k_2,\ell_2})^2 + \overline{h}_{k_2,\ell_2}^2} + \frac{b^2}{2} (c_{ij} - \widetilde{w}_{ij})^2 \ + \ F_0(\widetilde{u}),
\end{aligned} \tag{32}$$

where F_0 is not dependant on c_{ij} and

$$\overline{v}_{k_2,\ell_2} = \frac{v_{k_2,\ell_2} + h_{k_2,\ell_2}}{2}, \qquad \overline{h}_{k_2,\ell_2} = \frac{v_{k_2,\ell_2} - h_{k_2,\ell_2}}{2}. \tag{33}$$

Further we conclude that the local minimization problem for block (i,j) on level k is to minimize the functional

$$\begin{aligned}\overline{F}(c_{ij}) = \alpha\Big[&\sum_{\ell=\ell_1}^{\ell_2} \sqrt{(c_{ij} - h_{k_1-1,\ell})^2 + v^2_{k_1-1,\ell}} + \sum_{k=k_1}^{k_2-1} \sqrt{(c_{ij} - v_{k,\ell_2})^2 + h^2_{k,\ell_2}} + \\ &\sum_{\ell=\ell_1}^{\ell_2-1} \sqrt{(c_{ij} - h_{k_2,\ell})^2 + v^2_{k_2,\ell}} + \sum_{k=k_1}^{k_2} \sqrt{(c_{ij} - v_{k,\ell_1-1})^2 + v^2_{k,\ell_1-1}} + \\ &\sqrt{2}\sqrt{(c_{ij} - \overline{v}_{k_2,\ell_2})^2 + \overline{h}^2_{k_2,\ell_2}}\Big] + \frac{b^2}{2}(c_{ij} - \widetilde{w}_{ij})^2.\end{aligned} \tag{34}$$

The solution of (34) can be sought using the Richardson iterations as in (27).

As in 1D §3.3, it is equally feasible to formulate a varying patch coarse level as in the 2D case. The detection of a local patch (constant) can be done like in 1D by checking the relative differences. Let the general box (with general indices $k_1,\ k_2,\ \ell_1,\ \ell_2$) as depicted in (29). Then the local (varying) patch minimisation proceeds similarly to (32) and (34), with the only essential change in replacing b^2 (the old patch size) by $b_1 b_2$ (the new patch size); here $b_1 = k_2 - k_1 + 1$ and $b_2 = \ell_2 - \ell_1 + 1$. Here the patches are somewhat related to the active sets in other modified TV formulations [25, 8, 26].

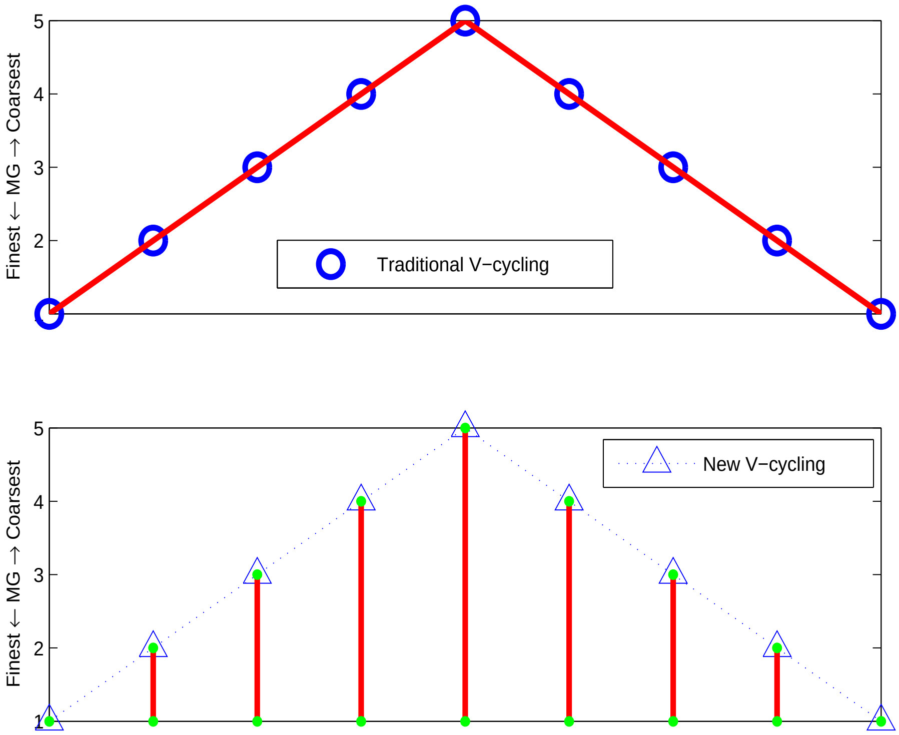
The 2D multilevel algorithm. We are ready to state our proposed piecewise constants based method.

Algorithm 2 *Given z and an initial guess $\widetilde{u} = z$, with $L+1$ levels,*

```
(1) Let $u_0 = \widetilde{u}$.
(2) Smooth the approximation on the finest level $1$, i.e. solve (26) for $j, \ell = 1, 2, \ldots, n$.
(3) On coarse levels $k = 2, 3, \ldots, L+1$:
    — compute $\widetilde{z} = z - \widetilde{u}$ via (31)
    — compute the local mean $\widetilde{w}_{ij} = mean\Big(\widetilde{z}\big((k-1)b+1 : kb, (\ell-1)b+1 : \ell b\big)\Big)$ via (31)
    — compute $\widetilde{v}_{k,\ell},\ \widetilde{h}_{k,\ell}$ via (33)
    — solve (34) by solving local minimisation as in (27) if $k \le L$ or
    — on the coarsest level $k = L+1$, the correction constant is simply
            $c = mean(\widetilde{w}) = mean(z - \widetilde{u})$.
    — Add the correction, $\widetilde{u} = \widetilde{u} + P_k c$ via (28).
(4) On level $k = 1$, check the possible patch size for each position $(i,j)$. Implement the piecewise
    constant update as with Step (3).
(5) If $\|\widetilde{u} - u_0\|_2$ is small enough, stop or return to Step (1).
```

We remark that although Algorithms 1 and 2 use the V-cycling like pattern in a multilevel setting, the precise manner of information transfers is different as illustrated by Figure 5. It is also feasible to develop other cycling patterns e.g. the W-cycling as described in [17]. In fact, some (though small) improvements have been observed with W-cycling.

Figure 5: V-cycling comparison. Top: the normal MG V-cycling and Bottom: the new V-cycling.



4 Numerical experiments

In this section, we shall test several aspects of the proposed multilevel methods. Firstly we demonstrate the convergence of the methods for several test cases and show that the proposed method gives fast and quality solutions in both 1D and 2D. Secondly we engage in the challenging task of comparing it to the well-known method of Chan-Golub-Mulet (CGM) [15] to show possible advantages of our parameter-free approaches. No methods could give better results than CGM as far as we know. It is thus particularly pleasing to present some test cases where we have achieved this.

Fast convergence. We first consider four 1D denoising problems with the signal-to-noise ratio (SNR) of 10 as shown in Figure 6. The processed solutions by our multilevel algorithm are shown in Figure 7, where one observes that the method is extremely fast (as expected of a multilevel method), converging to the tolerance of $tol = 10^{-4}$ in only 4 multilevel steps; here α takes the value of 7.68×10^{-5}, 19.2, 160, 25.6 respectively for the 4 examples. We then test further four examples of 2D denoising problems as shown in Figure 8, where SNR= 5. The computed solutions by our multilevel algorithm are shown in Figure 9. Again one observes that the method is extremely fast to obtain solutions which are comparable to from other methods. We next consider details of one comparison.

Comparison with CGM. As remarked, the method CGM [15] has been known to produce the most reliable solutions (in comparison to the fixed point and the time-marching methods). However CGM does depend on the parameter β; in particular if $\beta = 0$ (or nearly 0), CGM does not converge. We shall compare our MG algorithm to CGM with suitable β. Future comparisons should be made to a multilevel version of one of those recent methods that do not involve β (Section 2).

As an exact solution is not available, any comparison is attempting but difficult. What we propose to

Figure 6: Four 1D test examples

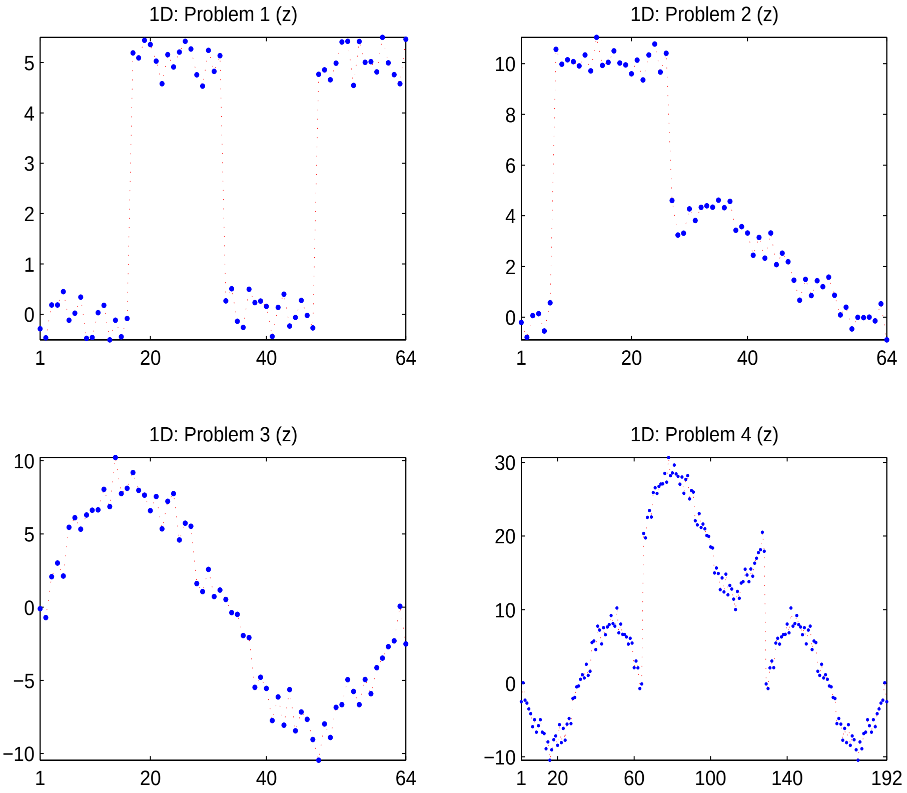


Figure 7: Primal multilevel method for the four 1D test examples

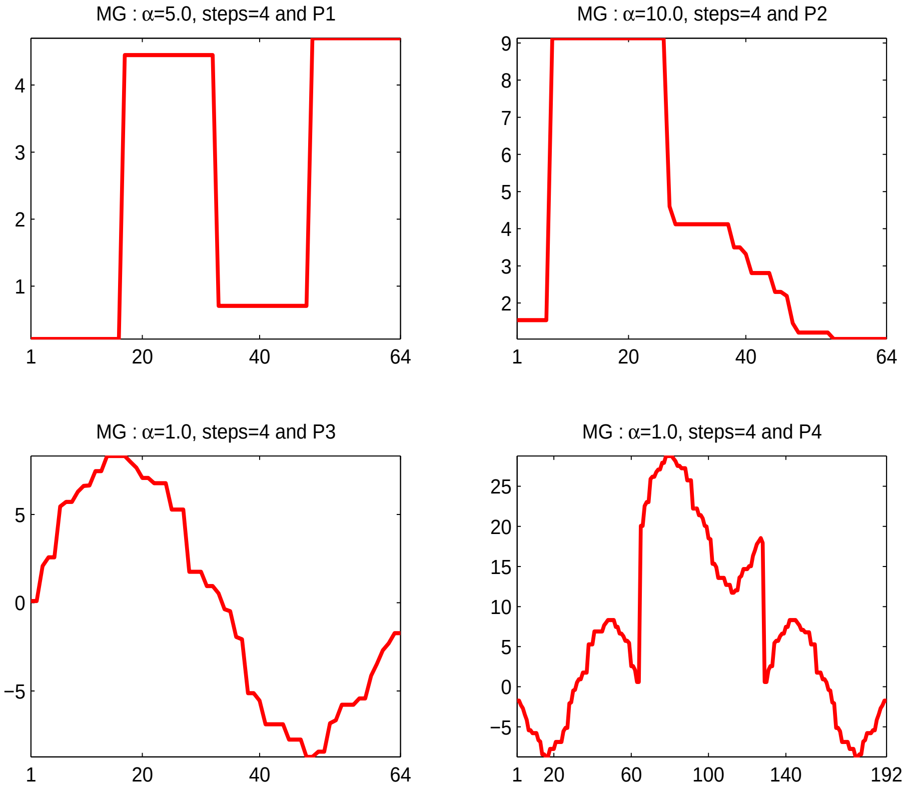


Figure 8: Four 2D test examples

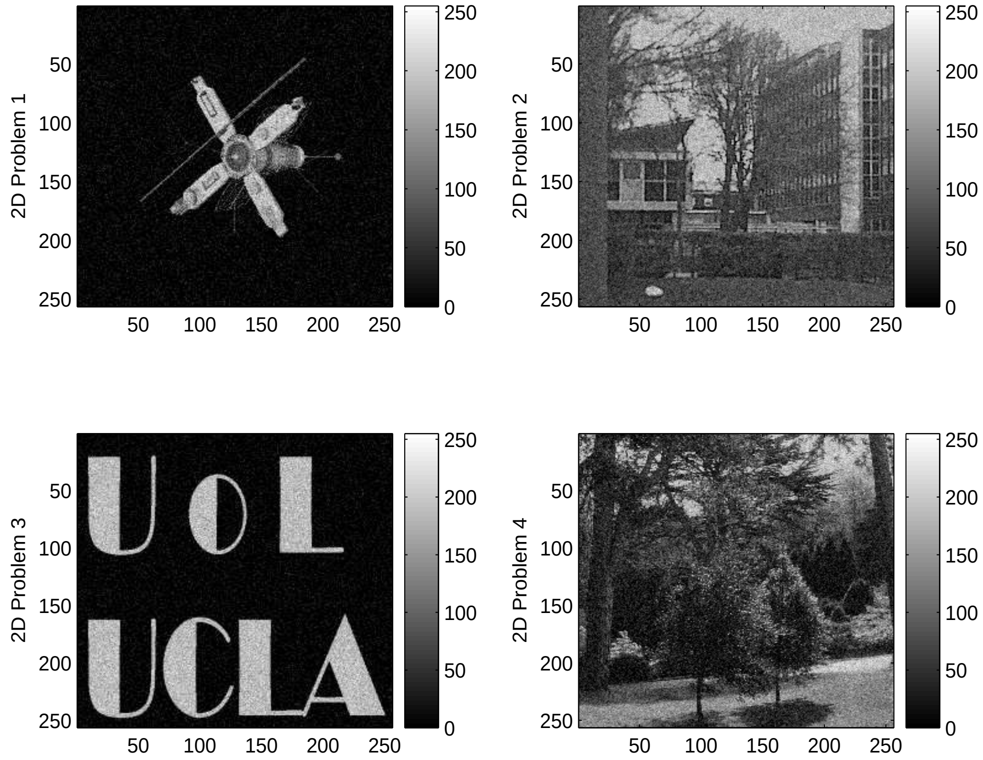


Figure 9: MG solutions of the 2D test examples

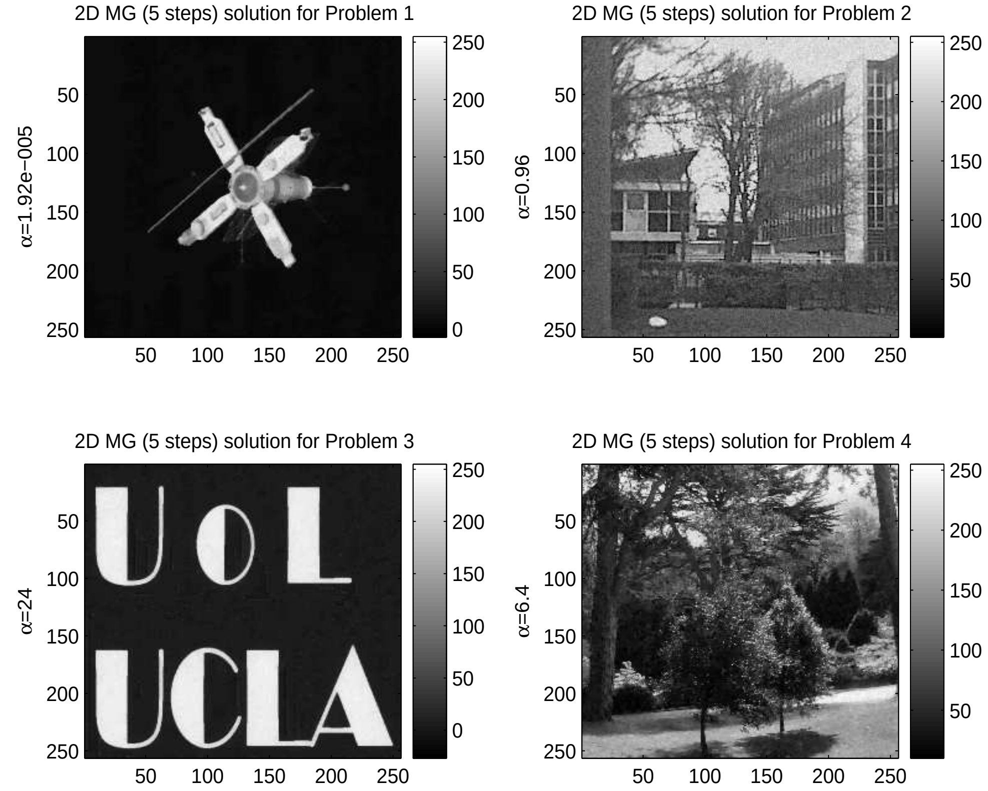

2D MG (5 steps) solution for Problem 1
2D MG (5 steps) solution for Problem 2
2D MG (5 steps) solution for Problem 3
2D MG (5 steps) solution for Problem 4
α=1.92e−005
α=0.96
α=24
α=6.4
50
100
150
200
250
0
50
100
150
200
250

Table 1: Comparison in merit function of our MGM with CGM [15] in 1D

Method	Problem	Convergence Steps	J(u)
CGM $\beta = 10^{-1}$	1	8	82.5160
	2	7	207.648
	3	6	34.1475
	4	6	135.143
CGM $\beta = 10^{-5}$	1	11	70.6723
	2	17	185.745
	3	11	32.8829
	4	11	131.433
CGM $\beta = 10^{-10}$	1	15	70.4687
	2	22	185.413
	3	13	32.8668
	4	11	131.389
CGM $\beta = 10^{-14}$	1	16	70.4680
	2	22	185.409
	3	13	32.8668
	4	11	131.390
MG (new)	1	2	70.4680
	2	4	185.407
	3	4	32.8664
	4	4	131.385

compare is CPU plus the value of the minimizing merit function $J(u)$ as in (13) for 1D. From the Table 1, one can see that our method is quite competitive to CGM in terms of reduction of $J(u)$ and CGM depends heavily on the choice of β as expected. For the 2D examples, we take the parameter $\gamma = \beta$ as we intend to compare the MG solutions with the CGM. For the image size 128 × 128 (for all the above 4 problems), we observe the results in Table 2. Clearly the parameter γ is less sensitive with the MG method than β with the CGM and of course it may be possible to replace the 2D local minimization by some parameter free methods e.g. [32] and then we expect the MG approach improves more.

Next we illustrate what the improvements over CGM solutions look like in terms of image sharpness. We shall take problem 2 (1D) as an example. With $\beta = 10^{-10}$, we have seen that $J(u_{CGM}) = 185.413$ while $J_{MG} = 185.407$ with the data size $n = 64$ (refer to the second plot of Figure 6). However, if we zoom in two locations of the solutions, we can see an interesting (but expected) difference between CGM and MG i.e. the MG solution is sharper (more like a staircase function) than the CGM. This is visible from Figure 10 near node 30 and Figure 11 near node 55 where we compare our MG solution (denoted by $\circ$) with 3 CGM solutions (denoted by $\bullet$) with respectively $\beta = \beta_1 = 10^{-4}, \beta_2 = 10^{-5}, \beta_3 = 10^{-6}$. Clearly the CGM solutions are smoother as expected (though almost piecewise constants).

Finally in Table 3, we display a brief comparison of CPU of our method with CGM. Although the problem sizes are small for a MG method, we can see that our method has shown the expected fast convergence.

5 Conclusions

This paper proposed a new and effective multilevel method based on piecewise constant refinements of the minimisation problem modelling the image denoising problem in the TV-norm. Our method is

Table 2: Comparison in merit function of our MGM with CGM [15] in 2D

Method	Problem	Convergence Steps	J(u)
CGM $\beta = 10^{-1}$	1	2	1.23384E-5
	2	7	6.59000E5
	3	10	2.55551E6
	4	10	4.31471E6
MG (new)	1	1	4.60572E-6
	2	3	6.59045E5
	3	5	2.67872E6
	4	5	4.31899E6
CGM $\beta = 10^{-5}$	1	3	1.17532E-5
	2	10	6.58996E5
	3	22	2.55007E6
	4	16	4.31415E6
MG (new) $\beta = 10^{-5}$	1	1	4.58596E-6
	2	5	6.59045E5
	3	11	2.70140E6
	4	8	4.31898E6
CGM $\beta = 10^{-10}$	1	7	4.50816E-6
	2	15	6.58996E5
	3	40	2.55002E6
	4	21	4.31414E6
MG (new) $\beta = 10^{-10}$	1	3	4.24560E-6
	2	7	6.59045E5
	3	20	2.68725E6
	4	10	4.31898E6

Table 3: Comparison in CPU of our MGM with CGM [15] in 2D (Problem 4 with $TOL = 10^{-6}$)

Size $N \times N$	MG (new)	CGM
64 × 64	11.2	11.9
128 × 128	44.7	48.9
256 × 256	207.3	2574.1

Figure 10: Detailed comparison of CGM and MG solutions near node 30 (Problem 2)

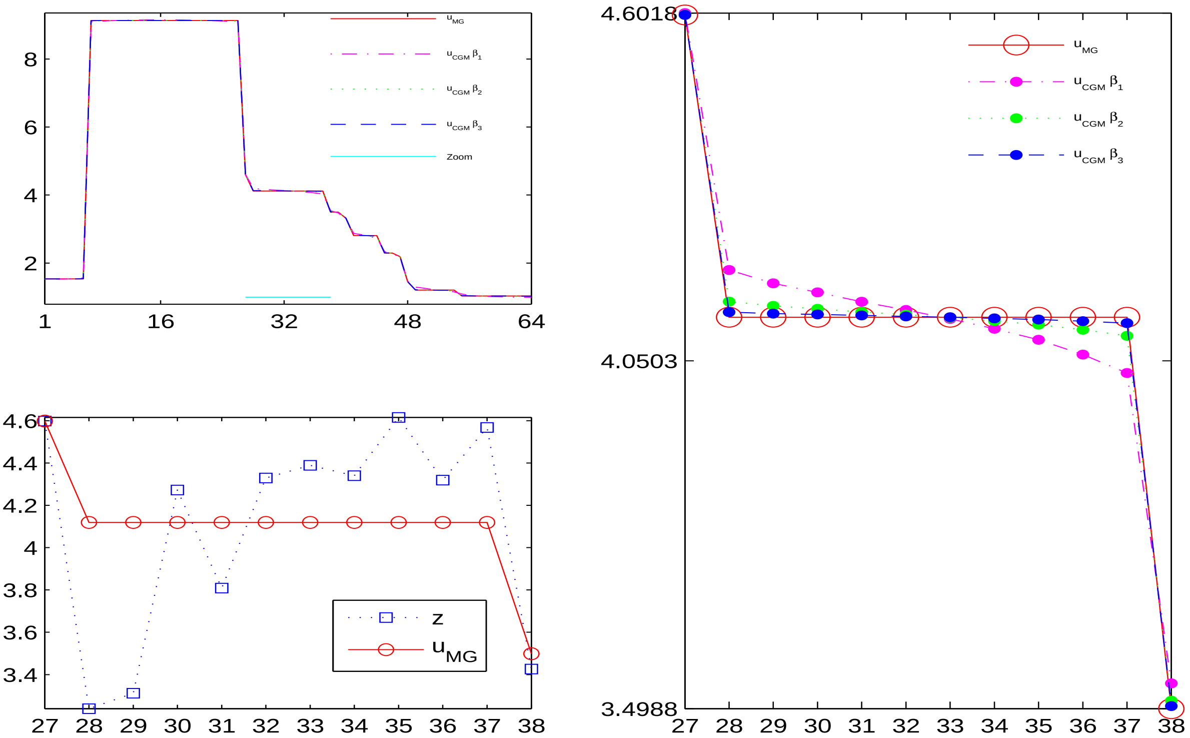
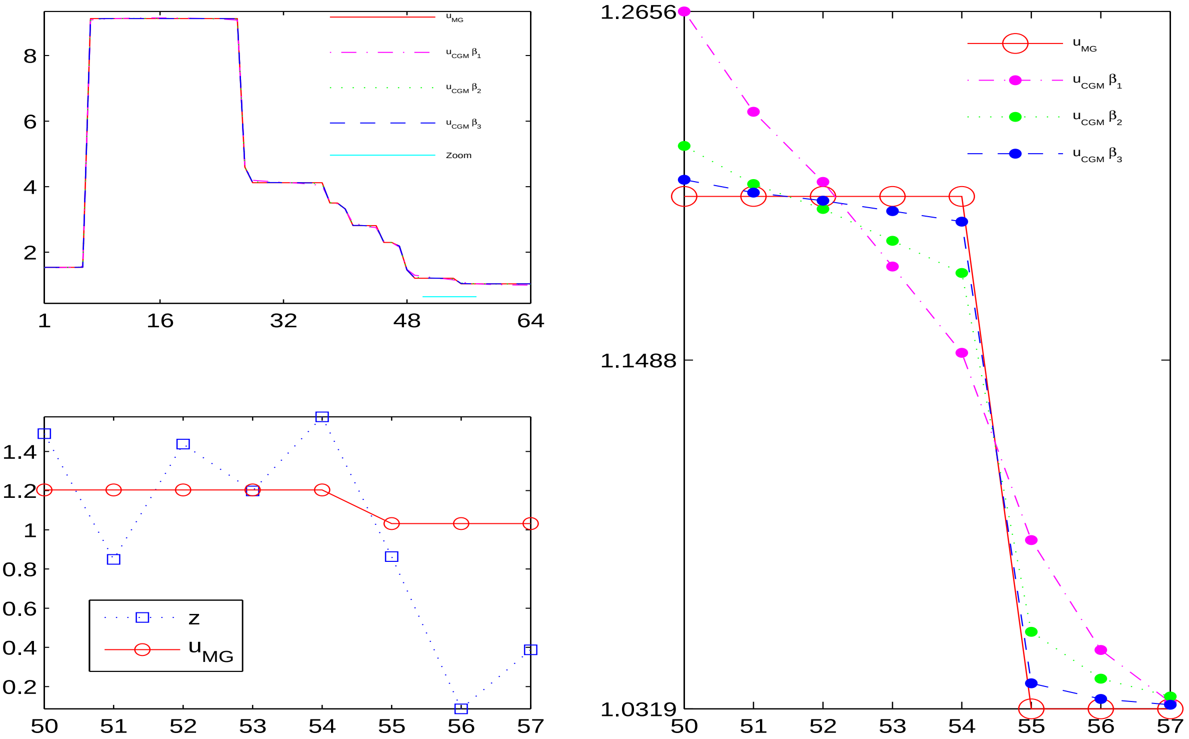


Figure 11: Detailed comparison of CGM and MG solutions near node 55 (Problem 2)



different from those multigrid methods applied to the PDEs and also from others which assume the minimising functional is differentiable. The apparently non-smooth solutions from primal relaxations of local minimisation are corrected using multilevels and varying size coarse level elements.

As our method is parameter free, as a by-product (when compared to the well-known CGM method), there is strong evidence to suggest that sharper solutions are obtained in many cases in the true spirit of TV norms. On the other hand, our method also works with variational models with a parameter ie. a smooth object functional. Of course, the main advantage of a multilevel method is its fast speed.

Acknowledgements

The authors thank the anonymous referees for making helpful remarks and suggestions. This work is supported in parts by the Office of Naval Research ONR N00014-03-1-0888, the National Institutes of Health NIH U54-RR021813 and the Leverhulme Trust RF/9/RFG/2005/0482.

References

[1] Acar R. and Vogel C. R. (1994), *Analysis of total variation penalty method for ill-posed problems*, Inverse Probs., 10, pp.1217-1229.

[2] Alvarez L., Lions P. L. and Morel J. M. (1992), *Image selective smoothing and edge detection by nonlinear diffusion II*, SIAM J. Numer. Anal., 29, pp. 845-866.

[3] Andrews H. C. and Hunt B. R. (1977), *Digital image restoration*, Prentice-Hall.

[4] Blomgren P., Chan T. F., Mulet P., Vese L. and Wan W. L. (2000), *Variational pde models and methods for image processing*, in: *Research Notes in Mathematics*, 420, pp. 43-67. Chapman & Hall/CRC.

[5] Bronstein M. M., Bronstein A. M., R. Kimmel R., and Yavneh I. (2006), *Multigrid multidimensional scaling*, Numer. Lin. Algebra Appl., to appear.

[6] Calvetti D., Lewis B. and Reichel L. (2001), *Krylov subspace iterative methods for nonsymmetric discrete ill-posed problems in image restoration*, Advanced Signal Processing Algorithms, Architectures, and Implementations X, ed. F.T. Luk, Proceedings of the Society of Photo-Optical Instrumentation Engineers (SPIE), vol. 4116, The International Society for Optical Engineering, Bellingham, WA, USA.

[7] Carter J. L. (2002), *Dual method for total variation-based image restoration*, CAM report 02-13, UCLA, USA; see http://www.math.ucla.edu/applied/cam/index.html

[8] Casas E., Kunisch K. and Pola C. (1999), *Regularization of functions of bounded variation and applications to image enhancement*, Appl. Math. Optim., 40, pp.229-257.

[9] Chambolle A. (2004), *An algorithm for total variation minimization and applications*, J. Math. Imaging Vis., 20, pp.89-97.

[10] Chambolle A. and Lions P. L. (1997), *Image recovery via total variation minimization and related problems*, Numer. Math., 76., pp.167-188.

[11] Chan R. H., Chan T. F. and Wan W. L. (1997), *Multigrid for differential convolution problems arising from image processing*, in: Proc. Sci. Comput. Workshop, Springer-Verlag, eds. R. Chan, T. F. Chan and G. H. Golub; see also CAM report 97-20, UCLA, USA.

[12] Chan R. H., Chan T. F. and Wong C. K (1999), *Cosine transform based preconditioners for total variation minimization problems in image processing*, IEEE Trans. Image Proc., 8, pp.1472-1478; see also CAM report 97-44, UCLA.

[13] Chan R. H., Chang Q. S. and Sun H. W. (1998), *Multigrid method for ill-conditioned symmetric toeplitz systems*, SIAM J. Sci. Comput., 19, pp. 516-529.

[14] Chan T. F. and Mulet P. (1996), *Iterative methods for total variation restoration*, CAM report 96-38, UCLA, USA; see http://www.math.ucla.edu/applied/cam/index.html

[15] Chan T. F., Golub G. H. and Mulet P. (1999), *A nonlinear primal dual method for total variation based image restoration*, SIAM J. Sci. Comput., 20, (6), pp. 1964-1977.

[16] Chang Q. S. and Chern I. L. (2003), *Acceleration methods for total variation-based image denoising*, SIAM J. Sci. Comput., 25, pp.982-994.

[17] Chen K. (2005), *Matrix Preconditioning Techniques and Applications*, Series: Cambridge Monographs on Applied and Computational Mathematics (No. 19), Cambridge University Press, UK.

[18] Goldfarb D. and Yin W. T. (2005), *Second-order cone programming methods for total variation-based image restoration*, SIAM J. Sci. Comput., 27 (2), pp.622-645.

[19] Gonzalez R. C. and Woods R. E. (1993), *Digital image processing*, Addison-Wesley.

[20] Henn S. and Witsch K. (1999), *A multigrid approach for minimizing a nonlinear functional for digital image matching*, Computing, 64 (4), pp.339-348.

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

[22] Hintermüller M. and Kunisch K. (2004), *Total bounded variation regularization as a bilaterally constrained optimization problem*, SIAM J. Appl. Math., 64, pp.1311-1333.

[23] Hintermüller and Stadler G. (2004), *An infeasible primal-dual algorithm for TV-based infconvolution- type image restoration*, Technical Report TR04-15, CAAM Dept. Rice University, USA.

[24] Hinterberger W., Hintermüller M., Kunisch K., von Oehsen M. and Scherzer O. (2003), *Tube Methods for BV Regularization*, J. Math. Imaging Vis. 19, pp.219-235.

[25] Ito K. and Kunisch K. (1999), *An active set strategy based on the augmented Lagrangian formulation for image restoration*, Math. Mod. Numer. Anal. (M2AN), 33 (1), pp.1-21.

[26] Kärkkäinen T. and Majava K. (2000), *Nonmonotone and monotone active set methods for image restoration II. numerical results*, J. Optim. Theory Appl., 106, pp.81-105.

[27] Kärkkäinen T., Majava K. and Mäkelä M. M. (2000), *Comparison of formulations and solution methods for image restoration problems*, Series B Report No. B 14/2000, Department of Mathematical Information Technology, University of Jyväskylä, Finland.

[28] Lee S. H. and Seo J. K. (2005), *Noise removal with Gauss curvature driven diffusion*, IEEE Trans. Image Proc., 14 (7), pp.904-909.

[29] Li Y. Y. and Santosa F. (1996), *A computational algorithm for minimizing total variation in image restoration*, IEEE Trans. Image Proc., 5, pp.987-995.

[30] Nash S. (2000), *A multigrid approach to discretized optimisation problems*, J. Opt. Methods Softw., 14, pp.99-116.

[31] Osher S. and Marquina A. (2000), *Explicit algorithms for a new time dependent model based on level set motion for nonlinear deblurring and noise removal*, SIAM J. Sci. Comput., 22(2), pp. 387-405.

[32] Powell M. J. D. (2004), *The NEWUOA software for unconstrained optimization without derivatives*, NA2004/08, Cambridge University, UK.

[33] Press W. H. et al (1992), *Numerical Recipes in C*, Cambridge University Press, UK.

[34] Radmoser E., Scherzer O. and Schöberl J. (2000), *A cascadic algorithm for bounded variation regularization*, SFB-Report No. 00-23, Johannes Kepler University of Linz, Austria.

[35] Riley K. L. (1999), *Two-Level Preconditioners for Regularized Ill-Posed Problems*, PhD thesis, Montana State University, USA.

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

[37] Sapiro G. (2001), *Geometrical differential equations and image analysis*, Cambridge University Press, UK.

[38] Savage J. and Chen K. (2005), *An improved and accelerated nonlinear multigrid method for total-variation denoising*, Int. J. Comput. Math., 82 (8), pp.1001-1015.

[39] Stuben K. (2001), *Algebraic Multigrid Methods*, In : U. Trottenberg, C. Oosterlee, A. Schuller, *Multigrid*, Academic Press, London, 2001.

[40] Vogel C. R. (1995), *A multigrid method for total variation-based image denoising*, In K. Bowers and J. Lund, eds., *Computation and Control IV*, 20, *Progress in Systems and Control Theory.* Birkhauser.

[41] Vogel C. R. (1999), *Negative results for multilevel precondtioners in image deblurring*, in: Scale-space theories in computer vision, pp. 292-304, eds. M. Nielson et al, Springer-Verlag.

[42] Vogel C. R. (2002), *Computational methods for inverse problems*, SIAM publications, USA.

[43] Vogel C. R. and Oman M. E. (1996), *Iterative methods for total variation denoising*, SIAM J. Sci. Statist. Comput., 17, pp. 227-238.

[44] Vogel C. R. and Oman M. E. (1998), *Fast, robust total variation-based reconstruction of noisy, blurred images*, IEEE Trans. Image Proc., 7, pp. 813-824.