J Sci Comput (2007) 33: 115–138
DOI 10.1007/s10915-007-9145-9

A Nonlinear Multigrid Method for Total Variation Minimization from Image Restoration

Ke Chen · Xue-Cheng Tai

Received: 2 September 2005 / Accepted: 22 June 2007 / Published online: 26 July 2007

Abstract Image restoration has been an active research topic and variational formulations are particularly effective in high quality recovery. Although there exist many modelling and theoretical results, available iterative solvers are not yet robust in solving such modeling equations. Recent attempts on developing optimisation multigrid methods have been based on first order conditions. Different from this idea, this paper proposes to use piecewise linear function spanned subspace correction to design a multilevel method for directly solving the total variation minimisation. Our method appears to be more robust than the primal-dual method (Chan et al., SIAM J. Sci. Comput. **20**(6), 1964–1977, 1999) previously found reliable. Supporting numerical results are presented.

Keywords Image restoration · Total variation · Regularisation · Subspace correction · Multilevel solvers

1 Introduction

Given a bounded domain $\Omega \subset \mathbb{R}^d$, $d = 1, 2, 3, \ldots$, we often need to solve problems which can be written in the following general form

$$\min_u \left(\int_\Omega |\nabla u| \, dx + \int_\Omega f(u) \, dx \right). \tag{1}$$

The equivalent problem to the above minimization is the Euler-Lagrange equation

$$-\nabla \cdot \left(\frac{\nabla u}{|\nabla u|} \right) + f'(u) = 0, \tag{2}$$

K. Chen (✉)
Department of Mathematical Sciences, University of Liverpool, Peach Street, Liverpool, L69 7ZL, UK
e-mail: k.chen@liverpool.ac.uk

X.-C. Tai
Department of Mathematics, University of Bergen, Bergen, Norway
e-mail: xue-cheng.tai@uib.no

which is a nonlinear partial differential equation (PDE), also known as a curvature equation [28]. The application of problems (1) and (2) ranges from image processing including noise removal [8, 25, 26, 31], segmentation [16, 22], deblurring [4], inverse problems [15] to interface motion driven by mean curvature [28, 30]. Owing to huge number of applications involved with models (1) and (2), the demand for new and fast solvers for these problems is equally huge. In this paper, we present a nonlinear multigrid method for efficiently solving (1).

In the literature the following methods have been used to solve (2):

(i) The fixed point iteration method [1, 40–44]: Once the coefficients $1/|\nabla \bar{u}|$ are fixed at a previous iteration $\bar{u}$, various iterative solver techniques have been considered [9–11, 24, 43, 44]. There exist excellent inner solvers but the outer solver can be slow. Further improvements are still useful.

(ii) The explicit time marching scheme [29, 31]: It turns the nonlinear PDE into a parabolic equation before using an explicit Euler method to march in time to convergence. The method is quite reliable but often slow.

(iii) The primal-dual (PD) method [4, 14, 17]: It solves for both the primal and dual variable together in order to achieve faster convergence with the Newton method (and a constrained optimisation with the dual variable). There does not appear to exist any multilevel version and also the inner solvers can have a convergence problem if the problem dimension gets large and β gets small.

As shown in [32], a converging multigrid method (MGM) can be much faster than methods of type (i) and (ii). In some cases, the MGM is also faster than the PD method (iii). The algorithms proposed in this paper behave similarly to [32] but, unlike [32], are not parameter dependent.

The MGM is one of the most powerful numerical methods for solving linear and nonlinear elliptic problems [38, 39, 46], although the method is known to be less robust for either case with highly discontinuous coefficients [45]. As for the curvature equation (2), several attempts have been made to develop MGM to solve it, cf. [2, 10, 32, 41]. However, the success so far is limited. The main problem is that the nonlinear diffusion coefficient $1/|\nabla u|$ can be highly oscillatory or degenerate (e.g. having large values close to infinity). Recently in [18, 23], the linear algebraic multigrid method [33] was adapted for solving the above PDE in each (outer) step of a fixed iteration while [32] attempted to use the standard multigrid methods with a non-standard and somewhat global smoother. As for solving (1) directly by MGM, the main obstacle to address is how to design the crucial coarse grid minimization problems for correction as no operator equations are directly available. Several related approaches, cf. [3, 5, 27, 34], tried to design such coarse grid problems by using first order conditions (similar to using (2) to measure residuals). However, the convergence of this kind of methods for certain nonlinear problems is not as good as for linear problems [18, 20, 23, 32]. Several authors [35–38] have studied the combined approach of MGM ideas and domain decomposition methods (DDM) (interpreted as subspace correction techniques) for some optimisation problems. Although it is proved that efficiency of DDM and MGM for a class of nonlinear problems is as good as for linear elliptic problems, there do not appear to exist any extensive uses of this approach for optimisation problems. Moreover, problem (1) is beyond the class of problems that were previously studied.

In this paper, we shall propose a nonlinear multigrid method for solving (1) based on subspace correction techniques. The essential idea is to use nonlinear smoothers for the subproblems which respect the minimization problem in order to reduce the energy functional. For the nonlinear problem (1), we shall demonstrate numerically that the efficiency of the

schemes can be as good as for linear problems. Thus we may summarise our contributions of this work as follows:

(a) We apply the subspace correction idea to design a nonlinear multigrid, as opposed to the geometric multigrid methods that were proposed [3, 5, 27, 34] based on regularising (2).
(b) The efficiency of the proposed algorithms is high: $O(N \log N)$ where N is the total degree of freedom.
(c) The proposed algorithms are not parameter dependent.
(d) As an inner-outer iteration procedure, our methods respect the nonlinear nature in the outer iteration in contrast to linearisation techniques.

The rest of the paper is organised as follows. In Sect. 2, we introduce the general subspace correction methods for convex functional minimisation. However we note that the theory does not cover the problem type (2). In Sect. 3, we detail our proposed multilevel algorithms for problem (2) and present some preliminary analysis. We present numerical experiments in Sect. 4 for solving both the one-dimensional and two-dimensional image denoising problems. Finally in Sect. 5, we discuss some conclusions and future work.

2 The Space Decomposition Algorithms

Consider a general minimization problem over a reflexive Banach space V:

$$\min_{v \in V} F(v), \tag{3}$$

where F is a strongly convex and Gateaux differentiable cost functional. Assume that the space V has been decomposed into a sum of smaller subspaces, i.e.

$$V = V_1 + V_2 + \cdots + V_m. \tag{4}$$

This means that for any $v \in V$, there exists $v_i \in V_i$ such that $v = \sum_{i=1}^{m} v_i$. Then the idea employed by [35–38] is to repeatedly solve the subspace minimisation of the type

$$\min_{v \in V_i} F(v^{(0)} + v),$$

where $v^{(0)}$ denotes a current approximation. Following [35–38], two types of subspace correction methods can be derived based on (4), namely the parallel subspace correction (PSC) method and the successive subspace correction (SSC) method, as simple generalisations of the methods for operator equations [46]. The parallel subspace correction method can be described as follows.

Algorithm 1 Choose an initial value $u^{(0)} \in V$ and relaxation parameters $\gamma_i > 0$ such that $\sum_{i=1}^{m} \gamma_i \leq 1$.

1. For $\ell \geq 0$, if $u^{(\ell)} \in V$ is defined, then find $p_i^{(\ell)} \in V_i$ in parallel for $i = 1, 2, \ldots, m$ such that

$$F\big(u^{(\ell)} + p_i^{(\ell)}\big) \leq F\big(u^{(\ell)} + v_i\big), \quad \forall v_i \in V_i. \tag{5}$$

2. Set

$$u^{(\ell+1)} = u^{(\ell)} + \sum_{i=1}^{m} \gamma_i p_i^{(\ell)}, \tag{6}$$

and go to the next iteration.

The successive subspace correction method can be described as follows:

Algorithm 2 Choose an initial value $u^{(0)} \in V$.

1. For $\ell \geq 0$, since $u^{(\ell)} \in V$ is defined, find $u^{(\ell+i/m)} = u^{(\ell+(i-1)/m)} + p_i^{(\ell)}$ with $p_i^{(\ell)} \in V_i$ sequentially for $i = 1, 2, \ldots, m$ such that

$$F\big(u^{(\ell+(i-1)/m)} + p_i^{(\ell)}\big) \leq F\big(u^{(\ell+(i-1)/m)} + v_i\big), \quad \forall v_i \in V_i. \tag{7}$$

2. Go to the next iteration.

The classical Gauss-Seidel and Jacobi relaxation methods and the modern DDM and MGM can all be interpreted as space decomposition algorithms. In order to reveal the relation between MGM and space decomposition, one can use finite element spaces. Similar explanations can also be given for finite difference approximations. For a given domain Ω, we assume that the finite element partition $\mathcal{T}$ of Ω is constructed by a successive refinement process. More precisely, $\mathcal{T} = \mathcal{T}_J$ for some $J > 1$, and $\mathcal{T}_j$ for $j \leq J$ are a nested sequence of quasi-uniform finite element partitions, i.e. $\mathcal{T}_j$ consist of finite elements $\mathcal{T}_j = \{\tau_j^i\}$ of size h_j such that $\Omega = \bigcup_i \tau_j^i$ for which the quasi-uniformity constants are independent of j and τ_{j-1}^l is a union of elements of $\{\tau_j^i\}$. We further assume that there is a constant $\gamma < 1$, independent of j, such that h_j is proportional to γ^{2j}. In Fig. 1 and Fig. 2, we plot the basis functions and the refined meshes for a domain in one and two dimensions. The basis functions associated with the boundary nodes are omitted in the 2D plot. For the two dimensional case, a finer grid is obtained by connecting the midpoints of the edges of the triangles of the coarser grid, with $\mathcal{T}_1$ being the given coarsest initial triangulation, which is quasi-uniform. In this example, $\gamma = 1/\sqrt{2}$. We can use much smaller γ in constructing the meshes, but the convergence will be slower. Corresponding to each finite element partition $\mathcal{T}_j$, a finite element space $\mathcal{M}_j$ can be defined by

$$\mathcal{M}_j = \{v : v|_\tau \in \mathcal{P}_1(\tau),\ \forall \tau \in \mathcal{T}_j\},$$

where $\mathcal{P}_1$ denotes the space of all piecewise linear elements (the basis functions are as illustrated in Fig. 1 and Fig. 2). Each finite element space $\mathcal{M}_j$ is associated with a nodal basis, denoted by $\{\phi_j^i\}_{i=1}^{n_j}$ satisfying

$$\phi_j^i\big(x_j^k\big) = \delta_{ik},$$

where $\{x_j^k\}_{k=1}^{n_j}$ is the set of all nodes of the elements of $\mathcal{T}_j$. Associated with each of such a nodal basis function, we define an one dimensional subspace as follows

$$V_j^k = \operatorname{span}\big(\phi_j^k\big).$$

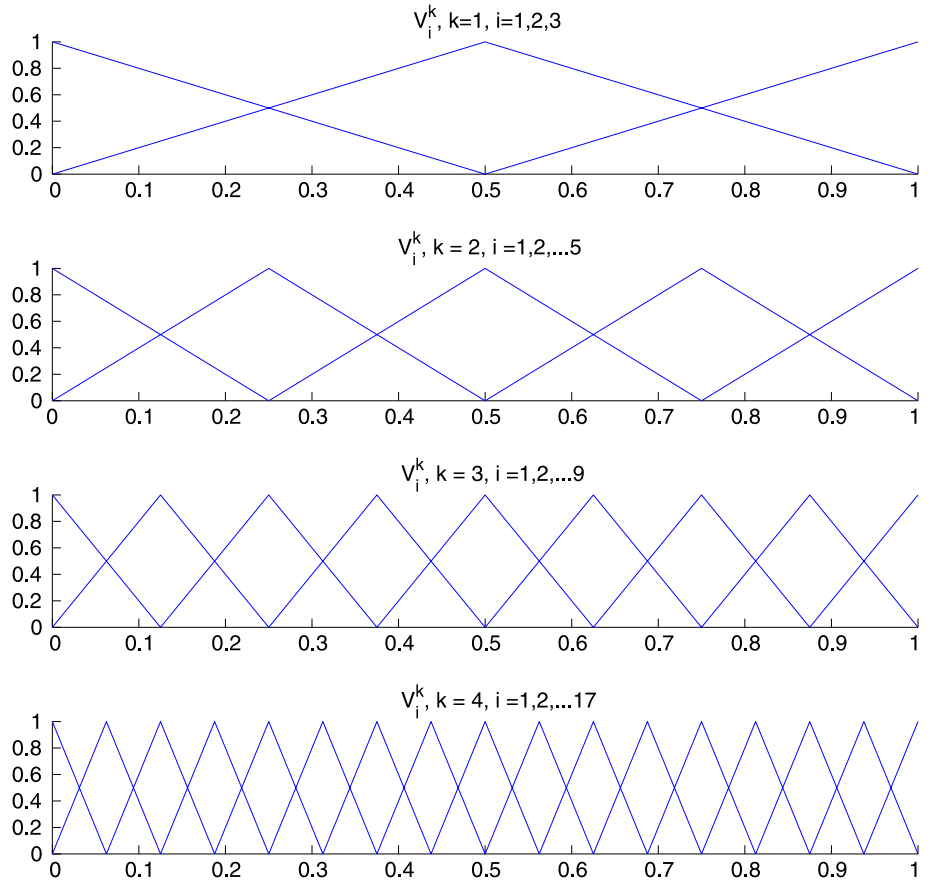

Fig. 1 Basis functions and the mesh for one dimensional multigrids

Letting $V = \mathcal{M}_J$, we have the following simple space decomposition:

$$V = \sum_{j=1}^{J} \sum_{k=1}^{n_j} V_j^k. \tag{8}$$

Each subspace V_j^k is one dimensional and thus the subproblem (7) is easy to solve.

For both of the algorithms, as the energy is monotonically decreasing during the iterations, it is easy to prove that solution $u^{(\ell)}$ converges to one minimizer of F under the condition that F is convex and continuous. If F is Lipschitz continuous and strongly convex, then an estimate on the rate of convergence can be obtained [35, 38]. Moreover, the convergence rate is mesh independent for the multigrid decomposition (8).

We remark that the above decomposition (8) is not an orthogonal (direct) sum in general. It has more basis functions than the hierarchical basis [19, 46]. Without multilevels, the simple decomposition $V = \sum_{k=1}^{n_1} V_1^k$ defined on the single level $j = 1$ (though leading to a relaxation method) is not suitable because it is known that relaxations alone cannot reach a global minimizer [6, 13]. It shall also be emphasized that this methodology differs

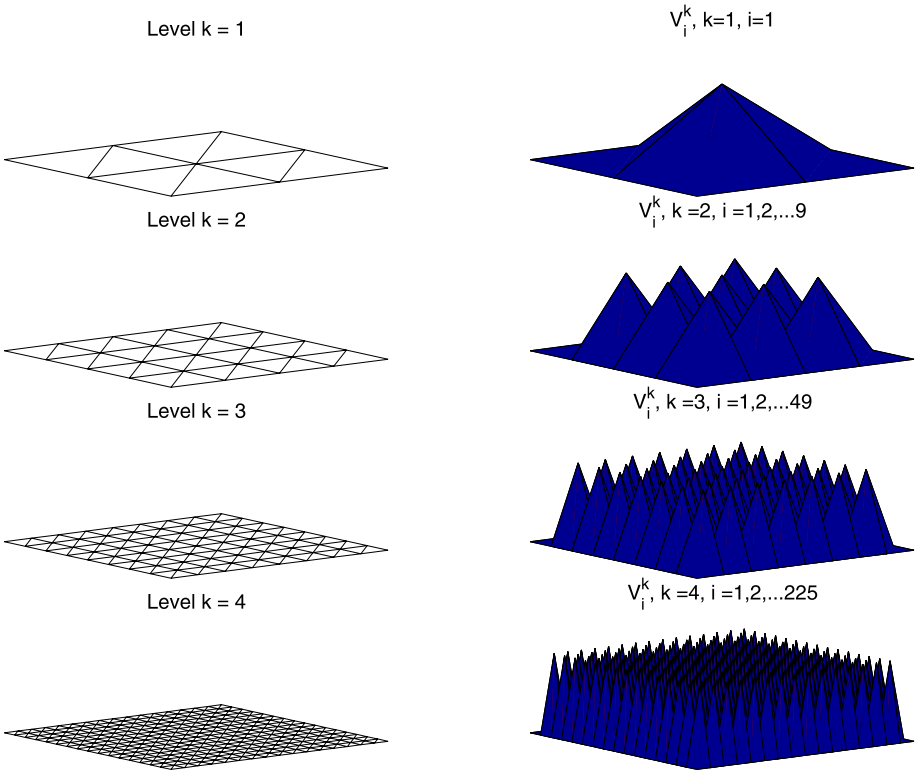

Fig. 2 Basis functions and the mesh for two dimensional multigrids

from [3, 5, 27, 34] on one major aspect: here we minimize the same functional on all levels while the other methods minimize a modified functional on coarse levels.

3 Image Restoration Algorithms Using the Total Variation Model

We now consider how to adapt the above methodology for solving (1).

For a given noisy image z defined on the domain $\Omega = [0, 1] \times [0, 1]$, one of the most well-known restoration models is the Rudin-Osher-Fatimi (ROF) total variation (TV) model [31] which is to take F in (3) to be

$$F(u) = \alpha \int_\Omega \sqrt{u_x^2 + u_y^2}\, dx + \frac{1}{2} \int_\Omega \|Ku - z\|^2\, dx, \tag{9}$$

where $\nabla u = (\frac{\partial u}{\partial x}, \frac{\partial u}{\partial y}) = (u_x, u_y)$ and K is a known operator. As the TV term $|\nabla u|$ is non-differentiable, one often replaces the above by a regularized functional

$$F(u) = \alpha \int_\Omega \sqrt{u_x^2 + u_y^2 + \beta}\, dx + \frac{1}{2} \int_\Omega \|Ku - z\|^2\, dx, \tag{10}$$

as done in [17, 32, 41] and almost all multilevel methods. In this paper, we are concerned with fast solution of this non-regularised model (9) in the denoising case with $K = I$. The

minimizer of (9) is taken as the denoised image. We note that a recent study [7] demonstrates that there are many advantages to transform (9) to a dual formulation; we expect to generalize our MGM to this dual model in the near future.

In the following, we shall use Algorithm 2 for problem (10). The functional F of (10) is convex and continuous in $BV(\Omega)$, the convergence of the algorithm is always guaranteed. In case that $\beta > 0$, then the functional F is strongly convex in $H^1(\Omega)$, thus the theory of [35, 38] guarantees that the algorithms has a convergence rate independent of the mesh size, but it will depend on β. In our numerical experiments, we have observed that the convergence rate is also independent of β. At the moment, it seems that there is still no analysis which can explain this fact.

In the following, we explain the details in using Algorithm 2 for multigrid decomposition (8). Note that all the subspaces in the multi-dimensional decomposition (8) are one dimensional. Thus, the subproblems (7) are essentially trying to solve the following one dimensional minimization problem:

$$\min_{c\in\mathbb{R}} F\big(w + c\phi_j^k\big), \tag{11}$$

where $w = u^{(\ell+(i-1)/m)} \in V$ and ϕ_j^k is the basis function over the jth level at the kth node. As F is convex, $c \in \mathbb{R}$ is a minimizer of (11) if and only if it satisfies

$$\int_\Omega \left[\alpha \frac{\nabla(w + c\phi_j^k)\cdot\nabla\phi_j^k}{\sqrt{|\nabla(w + c\phi_j^k)|^2 + \beta}} + \big(w + c\phi_j^k - z\big)\phi_j^k \right] dx = 0, \tag{12}$$

where one noticed that we have included a regularising parameter β for the local problem (11). Although we do not have to solve (11) this way, it turns out that the resulting method is not very sensitive to β unlike (10).

The key observation is that each of our local minimisation problems has only one degree of freedom (i.e. one dimensional). To solve this nonlinear equation for $c \in \mathbb{R}$, we may use the fixed point iteration (e.g. as in [1]), i.e. start with an $c^{(0)} = 0$ and recursively get $c^{(\ell)}$ from

$$\int_\Omega \left[\alpha \frac{\nabla(w + c^{(\ell+1)}\phi_j^k)\cdot\nabla\phi_j^k}{\sqrt{|\nabla(w + c^{(\ell)}\phi_j^k)|^2 + \beta}} + \big(w + c^{(\ell+1)}\phi_j^k - z\big)\phi_j^k \right] dx = 0. \tag{13}$$

It is easy to see that

$$c^{(\ell+1)} = \frac{b_j^k - a_j^k(w)}{a_j^k(\phi_j^k)}, \quad b_j^k = \int_\Omega (z - w)\phi_j^k\, dx \quad \text{and}$$

$$a_j^k(v) = \int_\Omega \left[\alpha \frac{\nabla v\cdot\nabla\phi_j^k}{\sqrt{|\nabla(w + c^{(\ell)}\phi_j^k)|^2 + \beta}} + v\phi_j^k \right] dx, \tag{14}$$

where $\ell \geq 0$ (global iteration), $j = 1, \ldots, J$ (all levels) and $k = 1, \ldots, n_j$ (level j). It is easy to see that $a_j^k(\phi_j^k) > 0$ so the iteration will not break down. As w is a function over the fine mesh, much reformulation will be done in integration for efficiently obtaining $a_j^k(w)$ and $a_j^k(\phi_j^k)$ as shown below. The iteration for (14) is stopped when $|c^{(\ell+1)} - c^{(\ell)}|/|c^{(\ell)}| \leq \tau_{\text{inner}}$. Numerical experiments will show that the convergence rate is nearly independent of τ_{inner}. Normally, just carrying out one or two iterations for (12) is sufficient to obtain required

results. Regarding complexity, we note that the domain integration in (13) and (14) does not present complications because the basis function ϕ_j^k is only defined locally (as with finite elements). This is addressed more precisely next.

3.1 The Algorithm with $\Omega \in \mathbb{R}$

Firstly we consider the case $\Omega \in \mathbb{R}$ associated with signal processing. Note that ϕ_j^k (on level $j = 2$ and at node $k = 2$) may be illustrated by Fig. 3. We wish to simplify the functional as much as possible by using the compact support of ϕ_j^k. As mentioned before, the parameter β is only introduced later for local minimization. (In the following, where boundary basis functions are involved, the usual adjustment in indices associated with summation is assumed.)

In the discrete setting for one dimensional problems, the cost functional (9) is (assuming α and F absorb the uniform step length $\Delta x = \Delta y = h$ from here onwards)

$$F(u) = \alpha \sum_{i=1}^{n-1} |D_x^+ u_i| + \frac{1}{2} \sum_{i=1}^{n} (u_i - z_i)^2,$$

where n is the total number of nodes, D_x^+ (also later D_y^+) is the standard forward finite difference operator. Let Ω_j^k be the support set of ϕ_j^k and $\bar{\Omega}_j^k$ be its closure. Corresponding to Ω_j^k, we define $I_j^k = \{i |\ x_i \in \Omega_j^k \cap \Theta\}$ and $\bar{I}_j^k = \{i |\ x_i \in \bar{\Omega}_j^k \cap \Theta\}$ with Θ being the set of the nodal points on the finest mesh. It is clear that we can localize the contribution of I_j^k

$$\begin{aligned} F(u) &= \left[\alpha \sum_{i \in [1,n] \setminus \bar{I}_j^k} |D_x^+ u_i| + \frac{1}{2} \sum_{i \in [1,n] \setminus I_j^k} (u_i - z_i)^2 \right] + \left[\alpha \sum_{i \in \bar{I}_j^k} |D_x^+ u_i| + \frac{1}{2} \sum_{i \in I_j^k} (u_i - z_i)^2 \right] \\ &= \widetilde{F_j^k}(u) + \alpha \sum_{i \in \bar{I}_j^k} |D_x^+ u_i| + \frac{1}{2} \sum_{i \in I_j^k} (u_i - z_i)^2, \end{aligned} \tag{15}$$

where $\widetilde{F_j^k}$ contains all terms not overlapping with the support of ϕ_j^k. Our task now is the following: given an initial guess $w \approx u$, how to improve w.

Our idea is to look for $u = w + c\phi_j^k$ for the best $c \in \mathbb{R}$. Recall that the above functional $F(u)$ is defined on the finest level so it is necessary to localize the formulation by collecting terms involving $c \in \mathbb{R}$ only (see (12)). Let $w = [w_1, \ldots, w_n]^T$ and $v_i = \phi_j^k(x_i)$. In the above functional (15), substitute u by $w + c\phi_j^k$ and then combine terms involving c:

$$\begin{aligned} F(w + c\phi_j^k) &= \widetilde{F_j^k}(w) + \alpha \sum_{i \in \bar{I}_j^k} |D_x^+ w_i + c D_x^+ v_i| + \frac{1}{2} \sum_{i \in I_j^k} (w_i - z_i + c v_i)^2 \\ &= \widetilde{F_j^k}(w) + \alpha \sum_{i \in \bar{I}_j^k} |D_x^+ w_i + c D_x^+ v_i| \ + \frac{1}{2} \left[s(c^2 - 2cz^* + z^{*2}) + \sum_{i \in I_j^k} \bar{z}_i^2 - z^{*2} s \right] \\ &= \overline{F_j^k}(w) + \alpha \sum_{i \in \bar{I}_j^k} |D_x^+ w_i + c D_x^+ v_i| \ + \frac{s}{2} (c - z^*)^2, \end{aligned} \tag{16}$$

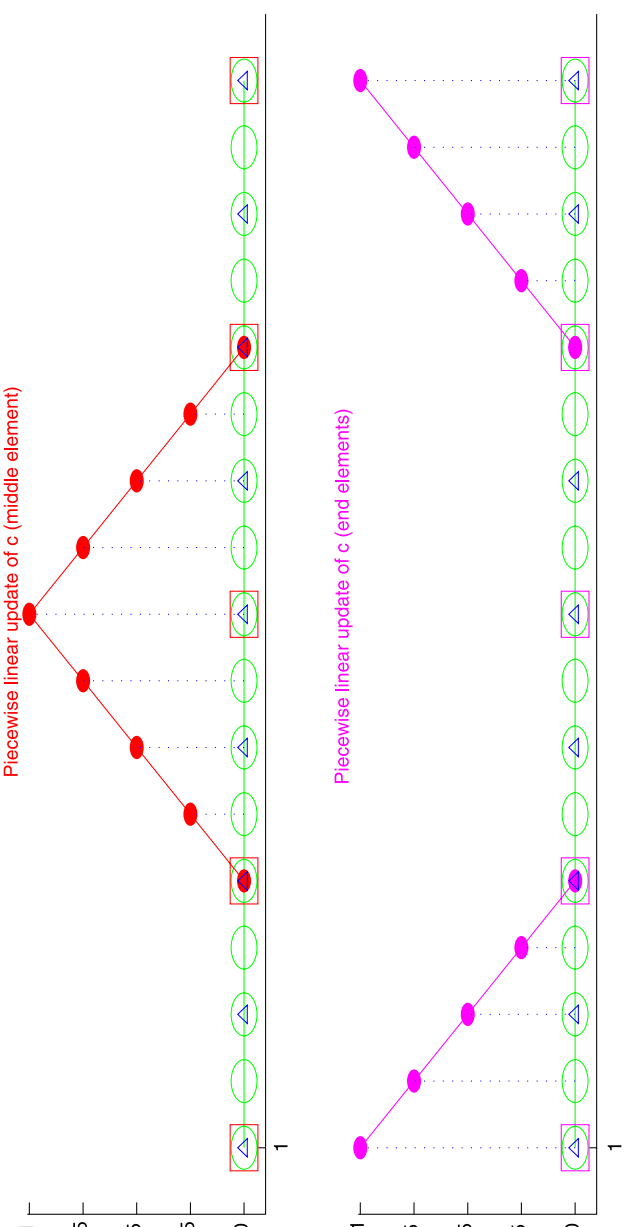

Fig. 3 The one dimensional basis function ϕ_j^k with • showing its height—on the coarse level $j = 2$ and at a middle node $k = 2$ (*top plot*) and at end nodes (*bottom plot*). Here ○ defines the finest level, △ refers to the first coarse level and □ to the second coarse level

where $\bar{z} = z - w$, $\overline{F_j^k}(w) = \widetilde{F_j^k}(w) + [\sum_{i\in I_j^k} \bar{z}_i^2 - z^{*2}s]/2$ does not involve c (ignored in subsequent minimisation),

$$s = \sum_{i\in I_j^k} v_i^2 \quad \text{and} \quad z^* = \sum_{i\in I_j^k} v_i \bar{z}_i / s.$$

Therefore in 1D, solving (11) for $c \in \mathbb{R}$ is equivalent to solving

$$\min_{c\in\mathbb{R}} \left[\alpha \sum_{i\in \bar{I}_j^k} |D_x^+ w_i + c D_x^+ v_i| \; + \frac{s}{2}(c - z^*)^2 \right],$$

and (with β added locally) the following

$$\min_{c\in\mathbb{R}} J(c), \qquad J(c) = \left[\alpha \sum_{i\in \bar{I}_j^k} \sqrt{(D_x^+ w_i + c D_x^+ v_i)^2 + \beta} + \frac{s}{2}(c - z^*)^2 \right]. \tag{17}$$

Further with $c^{(0)} = 0$, implementing (13) and (14) for (17) leads to the iterations

$$\left[\alpha \sum_{i\in \bar{I}_j^k} \frac{|D_x^+ v_i|^2}{\sqrt{(D_x^+ w_i + c^{(\ell)} D_x^+ v_i)^2 + \beta}} + s \right] c^{(\ell+1)}$$
$$= \left[s z^* - \alpha \sum_{i\in I_j^k} \frac{D_x^+ w_i \, D_x^+ v_i}{\sqrt{(D_x^+ w_i + c^{(\ell)} D_x^+ v_i)^2 + \beta}} \right], \quad \text{for } \ell = 0, 1, 2, \ldots. \tag{18}$$

In summary, our algorithm proceeds as follows.

Algorithm 3 Let the signal domain $\Omega = [0, 1]$ be discretized with J levels. Start from the finest level $j = 1$ with the initial guess $w = z$:

(1) On level j, compute $\bar{z} = z - w$ first.
(2) For each $k = 1, \ldots, n_j$:
First work out s and z^* and then solve the local coarse problem by iterating (18) until the relative (dynamic) residual is less than τ_{inner}.
Add the correction in the (built-in) interpolation step: $w = w + c\phi_j^k(x)$.
(3) If $j < J$, set $j := j + 1$ and continue with Step (1). If $j = J$, check whether the relative (dynamic) residual is less than τ_{outer}; if yes, exit with $u = w$ as the solution or otherwise continue with Step (1) with $j = 1$.

3.2 The Algorithm with $\Omega \in \mathbb{R}^2$

Secondly we can apply the same argument of simplification to the image case with $\Omega \in \mathbb{R}^2$, where we note that a 2D basis function ϕ_j^k (similar to Fig. 3) may be illustrated by Fig. 4. That is, the terms in the functional $F(w + c\phi_j^k)$, $c \in \mathbb{R}$, from (11) may again be grouped and simplified according to the compact support of ϕ_j^k. Similar to the 1D case in (16), the values

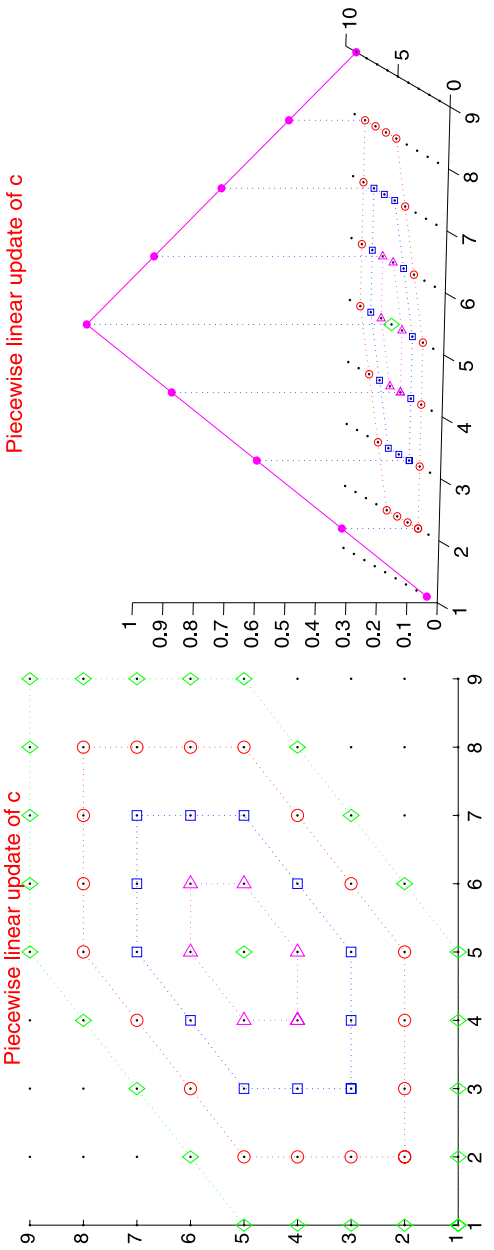

Fig. 4 The two dimensional basis function ϕ_j^k (on the coarse level $j = 3$ and at the center node k. Note on the *right plot*, only the weights v_ℓ along a diagonal, as in (19), are shown.) Here $\diamond$ defines the outer boundary of the 2D basis function, $\bigcirc$ shows the nodes where the corresponding weights are 1/4, $\square$ shows the nodes where the corresponding weights are 1/2, $\triangleright$ shows the nodes where the corresponding weights are 3/4 and the central node $\diamond$ defines the weight of 1

of the 2D basis function may be denoted by matrix v, which takes the values

$$v = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{4} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{4} & 0 & 0 & 0 \\ 0 & \frac{1}{4} & \frac{1}{2} & \frac{3}{4} & \frac{3}{4} & \frac{1}{2} & \frac{1}{4} & 0 & 0 \\ 0 & \frac{1}{4} & \frac{1}{2} & \frac{3}{4} & 1 & \frac{3}{4} & \frac{1}{2} & \frac{1}{4} & 0 \\ 0 & 0 & \frac{1}{4} & \frac{1}{2} & \frac{3}{4} & \frac{3}{4} & \frac{1}{2} & \frac{1}{4} & 0 \\ 0 & 0 & 0 & \frac{1}{4} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{4} & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{19}$$

for the example of $j = 3$ and $b = b_j = 4$ (as in Fig. 4) and when we zoom in the neighborhood of index k (as v is actually a global quantity with a compact support). Let the quantities v, Ω_j^k, I_j^k and $\bar{I}_j^k$ be defined in a similar way as for 1D problems with $k = (k_1, k_2)$. In the discretized setting, we have

$$\begin{aligned} F(u) &= \alpha \sum_{\ell_1=1}^{n} \sum_{\ell_2=1}^{m} \sqrt{(D_x^+ u_{\ell_1,\ell_2})^2 + (D_y^+ u_{\ell_1,\ell_2})^2} + \frac{1}{2} \sum_{\ell_1=1}^{n} \sum_{\ell_2=1}^{m} (u_{\ell_1,\ell_2} - z_{\ell_1,\ell_2})^2 \\ &= \widetilde{F_j^k}(u) + \alpha \sum_{k_1,k_2 \in \bar{I}_j^k} \sqrt{(D_x^+ u_{k_1,k_2})^2 + (D_y^+ u_{k_1,k_2})^2} + \frac{1}{2} \sum_{(k_1,k_2) \in I_j^k} (u_{k_1,k_2} - z_{k_1,k_2})^2, \end{aligned} \tag{20}$$

where $\widetilde{F_j^k}$ contains all terms not overlapping with the support of ϕ_j^k. Similar to the 1D case, we are ready to simplify $F(w + c\phi_j^k)$ to reveal the simplified minimisation for $c \in \mathbb{R}$ by grouping other unrelated terms (to c) into $\overline{F}$. The result is the following (refer to (15)):

$$\begin{aligned} &F(w + c\phi_j^k) \\ &\quad = F(w + cv) \\ &\quad = \widetilde{F_j^k}(w) + \alpha \sum_{(k_1,k_2) \in \bar{I}_j^k} \sqrt{(D_x^+ w_{k_1,k_2} + cD_x^+ v_{k_1,k_2})^2 + (D_y^+ w_{k_1,k_2} + cD_y^+ v_{k_1,k_2})^2} \\ &\qquad + \frac{1}{2} \sum_{(k_1,k_2) \in I_j^k} (\bar{z}_{k_1,k_2} - cv_{k_1,k_2})^2, \\ &\quad = \overline{F_j^k}(w, \bar{z}, v) + \alpha \sum_{(k_1,k_2) \in \bar{I}_j^k} T_{k_1,k_2}(c) + \frac{s}{2}(c - z^*)^2, \end{aligned} \tag{21}$$

where all three quantities $\widetilde{F}$, $\bar{z} = z - w$ and $\overline{F}$ do not involve c,

$$z^* = \sum_{(k_1,k_2) \in I_j^k} \frac{\bar{z}_{k_1,k_2} v_{k_1,k_2}}{s}, \quad s = \sum_{(k_1,k_2) \in I_j^k} v_{k_1,k_2}^2, \quad \text{and}$$

$$T_{k_1,k_2}(c) = \sqrt{|D_x^+(w_{k_1,k_2} + cv_{k_1,k_2})|^2 + |D_y^+(w_{k_1,k_2} + cv_{k_1,k_2})|^2}.$$

To solve the local problem (21), we re-define

$$T_{k_1,k_2}(c) = \sqrt{|D_x^+(w_{k_1,k_2} + cv_{k_1,k_2})|^2 + |D_y^+(w_{k_1,k_2} + cv_{k_1,k_2})|^2 + \beta},$$

adding a small parameter $\beta > 0$. Then omitting all non-essential details, we find that the updating of (14) or (13) in the discretized setting for a 2D problem proceeds as follows:

$$\left[\alpha \sum_{(k_1,k_2)\in \bar{I}_j^k} \frac{|D_x^+ v_{k_1,k_2}|^2 + |D_y^+ v_{k_1,k_2}|^2}{T_{k_1,k_2}(c^{(\ell)})} + s\right] c^{(\ell+1)}$$
$$= \left[sz^* - \alpha \sum_{(k_1,k_2)\in \bar{I}_j^k} \frac{D_x^+ w_{k_1,k_2} D_x^+ v_{k_1,k_2} + D_y^+ w_{k_1,k_2} D_y^+ v_{k_1,k_2}}{T_{k_1,k_2}(c^{(\ell)})}\right], \quad \text{for } \ell = 0, 1, 2, \ldots. \tag{22}$$

Putting all the steps together, we give the following

Algorithm 4 Let the image domain $\Omega = [0, 1] \times [0, 1]$ be discretized with J levels. Start from the finest level $j = 1$ with the initial guess $w = z$ over $n_1 \times m_1 = n \times m$ pixel points:

(1) On level j, compute $\bar{z} = z - w$ first.
(2) For each index $k = (k_1, k_2)$ with $k_1 = 1, \ldots, n_j$ and $k_2 = 1, \ldots, m_j$,
First work out s and z^* and then solve the local coarse problem by iterating (22) until the relative (dynamic) residual is less than τ_{inner}.
Add the correction in the (built-in) interpolation step: $w = w + c\phi_j^k(x)$.
(3) If $j < J$, set $j := j + 1$ and continue with Step (1). If $j = J$, check whether the relative (dynamic) residual is less than τ_{outer}; if yes, exit with $u = w$ as the solution or otherwise continue with Step (1) with $j = 1$.

Here, on level j, $n_j = (n-1)/2^{j-1} + 1$ and $m_j = (m-1)/2^{j-1} + 1$ define $n_j \times m_j$ basis functions.

Finally we briefly discuss the complexity issue. For linear problems, the cost per iteration for the multigrid iteration is typically $O(DOF)$ where DOF is the total number of degrees of freedom. For our nonlinear problems, the cost per iteration by our Algorithms 3 and 4 is $O(DOF \log(DOF))$. To verify this result, we may consider the 2D case with $DOF = N = mn$. Then the size of the set $\bar{I}_j^k$ is $2^{j-1} \times 2^{j-1}$ while the size of the set I_j^k is less than that of $\bar{I}_j^k$. Computing $\bar{z}$ requires N flops (floating point operations). For each k on level j, computing z^* and s requires $4b_j$ flops (with $b_j = 4^{j-1}$) so the total number of flops for level j is $4b_j n_j m_j \approx 4N$. Let t steps be needed for a typical inner iteration which corresponds to about $32b_j t n_j m_j \approx 32tN$. Hence over all J levels, the number of flops is $(N + 4N + 32tN)J = O(N \log N)$ since $\max(J) \le \log_2 \min(m, n) = O(N \log N)$.

4 Numerical Algorithms and Experiments

To demonstrate the effectiveness of our Algorithms 3 and 4, denoted by MG below, we now present some experimental results. We remark that the above proposed algorithms have not been applied to the image minimisation problem (2) before. It is pleasing to see some good results for the first time.

We shall first test the algorithms' effectiveness by solving a few image denoising problems in both 1D and 2D. Then we experiment on the dependence of the convergence of the proposed multigrid algorithm on image sizes and algorithm parameters. Finally we experiment on the influence of the inner Picard fixed point iterations (12) on the overall convergence performance. As we see, the method is not sensitive to the choice of problem sizes and accuracy of the inner Picard type fixed point iterations. Finally we compare and remark on the advantages of our algorithms over the popular method of [17]. In a simple word, our algorithms (being multilevel) are fast, robust and reliable.

4.1 Test Problems and Results

We shall consider 4 signal denoising problems as shown in Fig. 5 and another 4 image denoising problems as shown in Fig. 7. The signal-to-noise ratio (SNR) is taken as 10 (for smaller SNR all iterations will be less, as expected). The iterative method will be stopped whenever the relative dynamic residual $\|u^{(\ell)} - u^{(\ell-1)}\|_2/\|u^{(\ell)}\|_2 < \tau_{\text{outer}}$ for a prescribed tolerance τ_{outer}. Then ℓ will be the number of outer iteration steps (or *cycles*). There is another prescribed tolerance τ_{inner} which is to control how accuracy the iterations should be in the solution of the local minimisation (11). Here we take $\tau_{\text{inner}} = \tau_{\text{outer}} = 10^{-3}$ and $\beta = 10^{-4}$ for the regularising parameter. The processed results by our algorithm is shown in Fig. 6 (for $N = 4097$) and Fig. 8 (for $N \times N = 257 \times 257$) respectively, where the symbol $\square$ refers to our algorithm while the symbol $\times$ the method of [17]. Clearly one observes that our method converges quite quickly and gives a result which is not distinguishable from the result of [17].

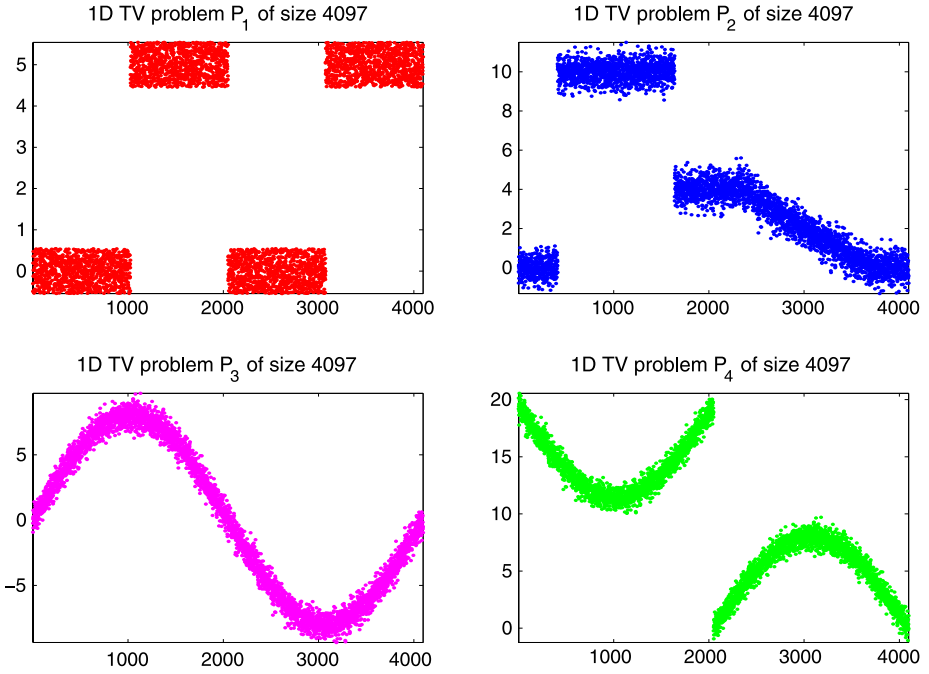

Fig. 5 The 1D test examples

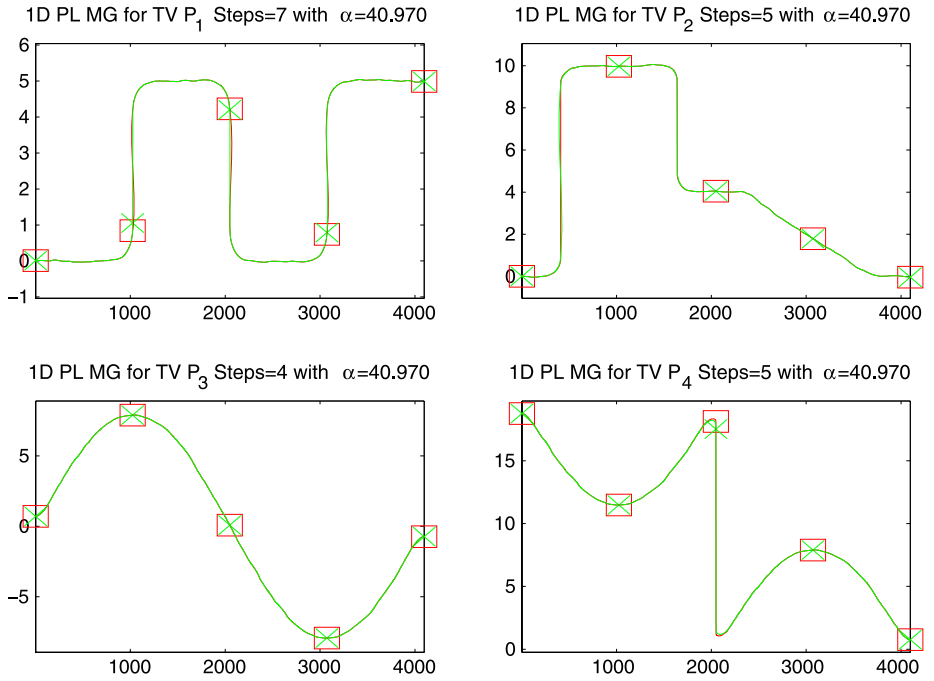

Fig. 6 The 1D processed results with solutions from the two methods superimposed on each other: □—the new multilevel algorithm and ×—the primal-dual method [17]

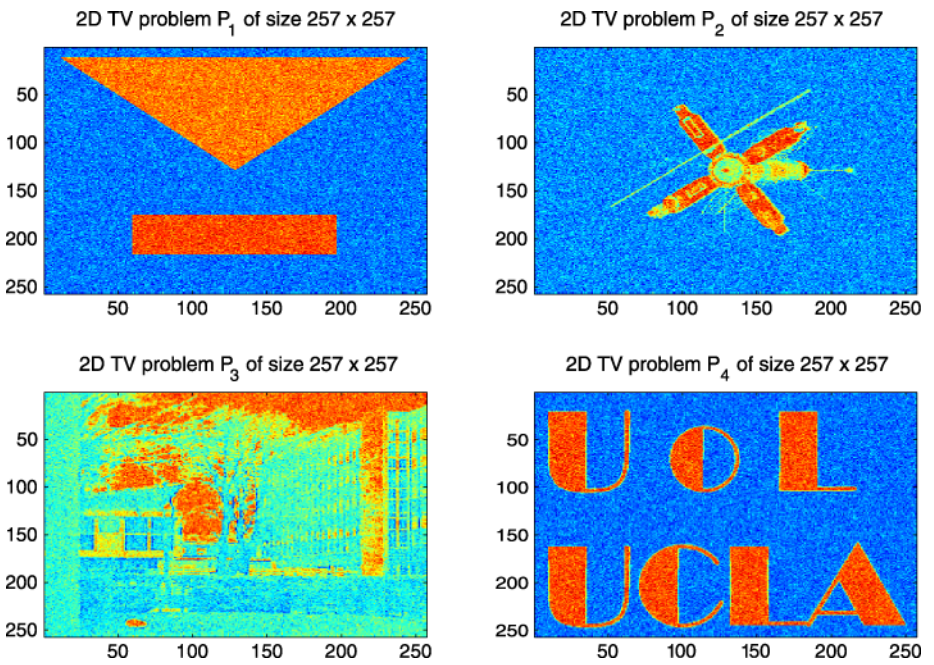

Fig. 7 The 2D test examples

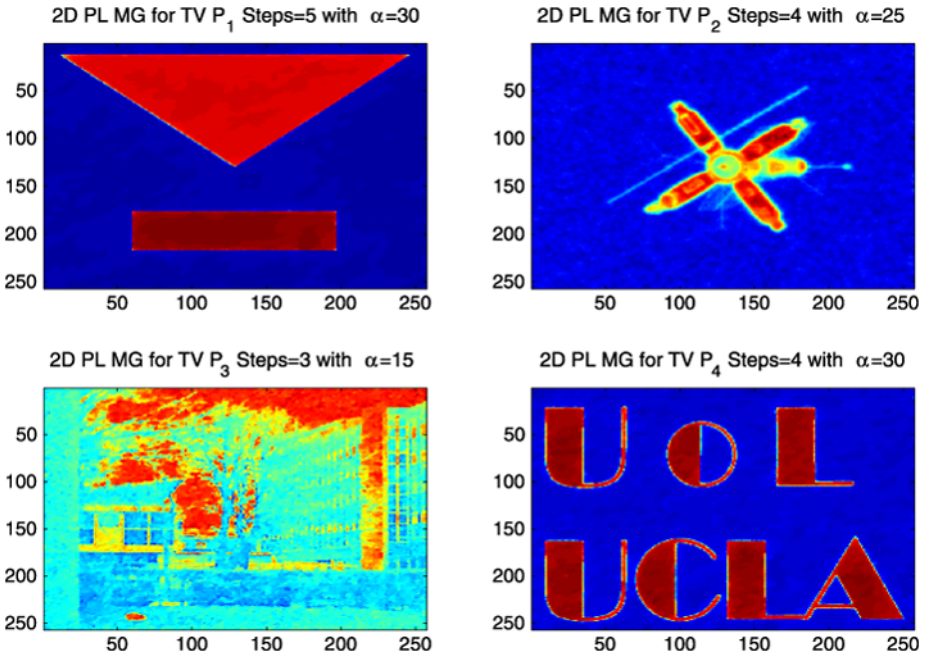

Fig. 8 The 2D multigrid restored results

4.2 Test of Convergence of the Method

When $\beta \neq 0$, the convergence theory developed in [35, 38] may be invoked to establish a convergence result of our algorithms. Here we hope to give some numerical tests to demonstrate the convergence behaviour for the specific example of problem 3 with $\alpha = 15$. First we show in Fig. 9 the convergence history of MG residuals for $\tau = \tau_{\text{outer}} = 10^{-10}$ and $n = 129$. Second we show some more residual information in Table 1 for various n with up to 100 steps of the MG method to achieve $\|r\| \leq \tau = 10^{-10}$. Clearly convergence slows downs at we approach the machine accuracy but it is not much depending on n. Hence in the following tests, we shall restrict ourselves to a larger τ.

To test further on the sensitivity of the method on problem size n, it is of interest to investigate any dependence of the overall algorithm convergence as the problem sizes increase (n in 1D signals and $n \times n$ in 2D images). In Table 2, we fix both tolerances $\tau_{\text{inner}}, \tau_{\text{outer}}$ and vary the problem size to see how many convergence steps are needed. Clearly one observes that the convergence of our method is not much affected by n, especially for the 2D problems. For the 1D case, the convergence patterns become clear and the number of steps (i.e. *MG cycles*) approaches a constant as n increases.

4.3 Sensitivity to the Inner Fixed Point Iterations

We next address how crucial the inner nodal solver is for the overall algorithm. To this end, we fix the problem size n and the tolerance τ_{outer}. Table 3 shows the results obtained for the selected test problems in 1D and 2D from varying the inner solver tolerance τ_{inner} within the range of a value below τ_{outer} to another much larger value. Clearly the overall multilevel method is not much affected. Note that for the cases associated with using the

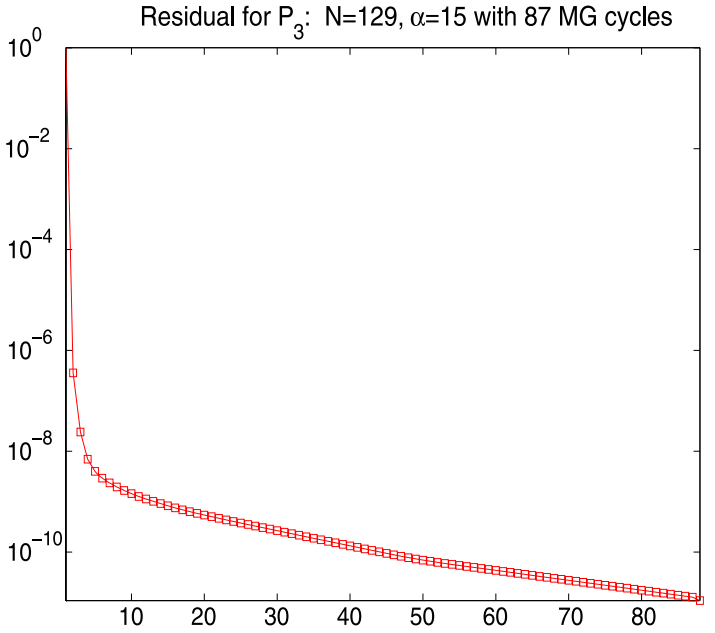

Fig. 9 Residual history for problem 3 with $n = 129$

Table 1 Residual information for problem 3 with up to 100 MG cycles

Problem size	Levels	MG cycles	Residual
17×17	5	87	8.5E-10
33×33	6	100	5.9E-10
65×65	7	93	5.0E-10
129×129	8	87	1.1E-11
257×257	9	77	2.2E-11

largest tolerance $\tau_{\text{inner}} = 10$, the number of inner iterations is mostly one and hence the inner solver is far from convergence and yet the outer iterations can converge. This latter observation is somewhat related to the inner-outer iteration control as shown in [21] and adopted in the algorithm of [17]. It is possible to work out an appropriate formula for τ_{inner}.

4.4 Sensitivity of the Parameters α and β

There are two general issues here. Firstly one cares about whether or not α and β affect the convergence of a method. Secondly for difficult choices of α and β, one desires for a remedial solution. Here we mainly test the former as our algorithms are not sensitive to such parameter changes. As for the latter question with other sensitive methods, one should consider the parameter continuation idea as used and discussed in [18, 47].

We take two test examples as shown in Figs. 10 and 11. We have done the following experiments (for the tolerance of $\tau = 10^{-3}$) in Table 4.

Table 2 Test of dependence of the problem sizes (n in 1D and $n \times n$ in 2D): 'Dim' denotes 'Dimension', 'Prob' stands for 'Problem number', 'Levels' indicates the "levels used in the multilevel algorithm" and 'Steps' the "number of multilevel cycles". Here $\tau = 10^{-3}$, $\beta = 10^{-4}$. Clearly there is no strong dependence. Here the problem numbers refer to Fig. 5 for 1D and Fig. 7 for 2D

Dim	Prob	Size	Levels	Steps	Prob	Size	Levels	Steps
1D	1	65	6	25	2	65	6	17
		129	7	11		129	7	11
		257	8	7		257	8	5
		513	9	8		513	9	6
		1015	10	5		1015	10	4
		2049	11	4		2049	11	4
		4097	12	3		4097	12	4
		8193	13	4		8193	13	4
		16385	14	3		16385	14	4
		32769	15	3		32769	15	4
		65537	16	3		65537	16	4
1D	3	65	6	9	4	65	6	34
		129	7	7		129	7	23
		257	8	8		257	8	18
		513	9	5		513	9	12
		1015	10	5		1015	10	8
		2049	11	5		2049	11	6
		4097	12	5		4097	12	4
		8193	13	4		8193	13	4
		16385	14	4		16385	14	4
		32769	15	4		32769	15	4
		65537	16	4		65537	16	4
2D	1	33×33	5	6	2	33×33	5	5
		65×65	6	6		65×65	6	5
		129×129	7	6		129×129	7	5
		257×257	8	6		257×257	8	5
2D	3	33×33	5	6	4	33×33	5	5
		65×65	6	6		65×65	6	5
		129×129	7	6		129×129	7	5
		257×257	8	6		257×257	8	5

Here we measure the restoration qualitatively by the peak signal-to-noise ratio (PSNR) defined by (see e.g. [12])

$$\mathrm{PSNR}(u, w) = 10 \log_{10} \frac{255^2}{\frac{1}{mn} \sum_{i,j} (u_{i,j} - w_{i,j})^2},$$

where $w_{i,j}$ and $u_{i,j}$ denote the pixel values of the restored and the original images respectively.

Table 3 Test of dependence of the accuracy of the inner nodal solver ($n = 8193$ and levels $= 13$ in 1D and $n \times n = 257 \times 257$ and levels $= 8$ in 2D): 'levels' indicates the "levels used in the multilevel algorithm" and 'steps' the "number of multilevel cycles". Here $\tau = \beta = 10^{-4}$ and τ_{inner} is the tolerance used for each nodal relaxation solver (note: the minimal number of relaxation steps is 1). Clearly there is no strong dependence. Here again, the problem numbers refer to Fig. 5 for 1D and Fig. 7 for 2D

Dimension	Problem	τ_{inner}	Steps	Problem	τ_{inner}	Steps
1D	1	10^{-5}	8	2	10^{-5}	4
		10^{-4}	8		10^{-4}	4
		10^{-3}	8		10^{-3}	4
		10^{-2}	8		10^{-2}	4
		10^{-1}	8		10^{-1}	4
		10^{-0}	8		10^{-0}	4
		10^{+1}	8		10^{+1}	4
1D	3	10^{-5}	5	4	10^{-5}	11
		10^{-4}	5		10^{-4}	11
		10^{-3}	5		10^{-3}	11
		10^{-2}	5		10^{-2}	11
		10^{-1}	5		10^{-1}	11
		10^{-0}	5		10^{-0}	11
		10^{+1}	5		10^{+1}	11
2D	1	10^{-5}	10	2	10^{-5}	5
		10^{-4}	10		10^{-4}	5
		10^{-3}	10		10^{-3}	5
		10^{-2}	8		10^{-2}	5
		10^{-1}	8		10^{-1}	5
		10^{-0}	10		10^{-0}	5
		10^{+1}	11		10^{+1}	5
2D	3	10^{-5}	6	4	10^{-5}	7
		10^{-4}	6		10^{-4}	7
		10^{-3}	6		10^{-3}	7
		10^{-2}	6		10^{-2}	7
		10^{-1}	6		10^{-1}	7
		10^{-0}	6		10^{-0}	8
		10^{+1}	6		10^{+1}	9

Clearly from Table 4, we observe that the convergence of Algorithm 4 is not significantly affected by parameter changes. Evidently changing α leads to different restorations and hence the PSNR values as expected.

One may wonder why our optimisation MG is less sensitive to the above parameters while the MG is more sensitive for the essentially same model (1). We believe that this is due to the PDE (2) attempting to assign a normal at pixels where such geometrical information is not defined while the optimisation does not require such assignments.

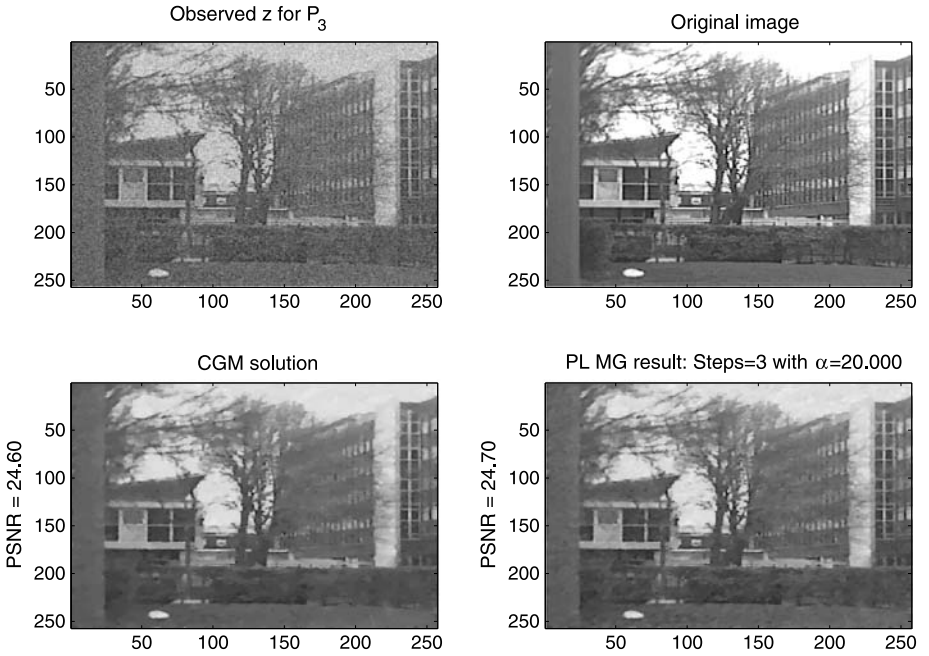

Fig. 10 Comparison with the CGM method [17] for test example P_3: $\alpha = 20$ and $\beta = 10^{-20}$

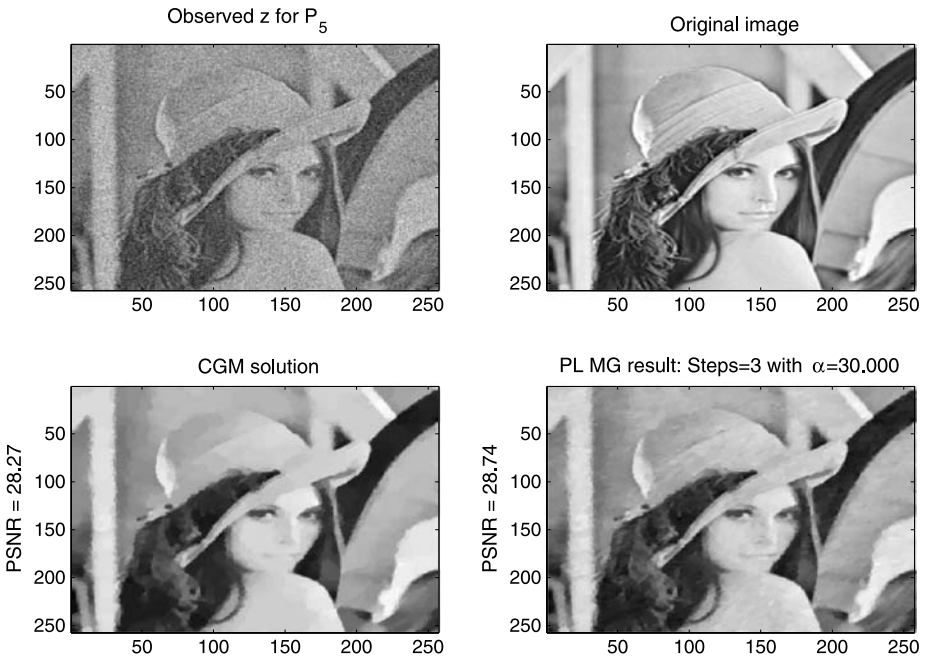

Fig. 11 Comparison with the CGM method [17] for test example P_5: $\alpha = 30$ and $\beta = 10^{-20}$

Table 4 Test of dependence of the parameters α and β

Problem	α	β	MGM cycles	PSNR
3	1.25	10^{-12}	6	20.67
	2.50		6	21.31
	5.00		6	22.53
	10.0		6	24.38
	20.0		8	24.68
3	20.0	10^{-4}	8	24.68
		10^{-8}	8	24.68
		10^{-12}	8	24.68
		10^{-16}	8	24.68
		10^{-20}	8	24.68
5	1.88	10^{-12}	6	21.12
	3.75		6	22.26
	7.50		6	24.56
	15.0		6	27.94
	30.0		9	28.75
5	30.0	10^{-4}	8	28.69
		10^{-8}	8	28.70
		10^{-12}	8	28.70
		10^{-16}	8	28.70
		10^{-20}	8	28.71

4.5 Comparisons with an Established Method

There are many aspects that could be compared with other methods. Here we choose to compare with the well-known method (perhaps the best) of Chan-Golub-Mulet (CGM) [17] as other methods such as the fixed point iterations and time marching schemes have been shown to be slower than a multigrid method [32]. However our task of comparing with CGM becomes somewhat easier because the CGM method 'fails' in 2 cases: (i) when the image size N becomes large (due to ill-conditioning); (ii) when $\beta \leq 10^{-32}$ (due to singularity). Here (i), not (ii), may be fixable but no such work is available for the primal-dual method. (However there exists important work of β-free methods [1, 7, 13]; of these the dual method is the most well-known.) In either of these cases, our method would converge although the local solvers take a few more iterations.

It may be of interest to show some results from parameter ranges where the CGM performs well: we take $\beta = 10^{-20}$ and 2 test examples in Figs. 10 and 11. Here we mainly compare the solution's visual quality and the PSNR values. As seen from Figs. 10 and 11, $\text{PSNR}(u_{CGM}) = 24.60$ and $\text{PSNR}(u_{MG}) = 24.70$ for problem 3, and $\text{PSNR}(u_{CGM}) = 28.27$ and $\text{PSNR}(u_{MG}) = 28.74$ for problem 4. Since the PSNR values of the results from our algorithm are quite close to the CGM results, the restored images are indeed indistinguishable.

For larger images, our MG method can solve problems 3 and 4 in a reasonable time (as seen below) on a Sun-Blade 1000 with Matlab 6.5:

Problem	N	MGM cycles	PSNR	CPU
3	513	4	31.48	3165.6
	1025	4	34.03	14478.0
4	513	4	30.16	4291.0
	1025	4	33.09	14428.0

In contrast, CGM cannot be run because the memory requirement is too large.

Therefore our algorithm is evidently more robust (without having to concern about what parameters to use) and being a multilevel method there is a scope to achieve even better performance with future parallelization.

5 Conclusions

This paper has introduced a nonlinear multigrid method for solving curvature equations related to total variation minimization. The resulting algorithms are efficient and different from the existing optimisation multigrid methods in coarse level construction.

Mesh independent convergence has been observed from the numerical experiments. Moreover, the iteration number needed to achieve a given accuracy for the solution also seems to be independent of β. The same is also true for a large range of α. A theoretical analysis for this behavior is still missing.

It is known that the CGM algorithm [17] is rather robust with respect to α and β. For most of the experiments we have done, we need less than 10 outer iterations to get a result as nearly undistinguishable as the unigrid method [17]. In fact, the CGM method fails to converge when we take $\beta = 10^{-32}$ while our method converges. If we must take $\beta = 0$, all we need to do is to replace our local minimisation solvers.

Acknowledgements The first author wishes to thank the support of the Leverhulme Trust RF/9/RFG/2005 /0482 for this work.

References

1. Acar, R., Vogel, C.R.: Analysis of total variation penalty method for ill-posed problems. Inverse Probl. **10**, 1217–1229 (1994)
2. Acton, S.T.: Multigrid anisotropic diffusion. IEEE Trans. Image Process. **3**(3), 280–291 (1998)
3. Arian, E., Ta'asan, S.: Multigrid one-shot methods for optimal control problems. ICASE Technical Report No. 94-52, USA (1994)
4. Blomgren, P., Chan, T.F., Mulet, P., Vese, L., Wan, W.L.: Variational PDE models and methods for image processing. In: Research Notes in Mathematics, vol. 420, pp. 43–67. Chapman & Hall/CRC (2000)
5. Brandt, A.: Multigrid solvers and multilevel optimization strategies. In: Cong, J., Shinnerl, J.R. (eds.) Multiscale Optimization and VLSI/CAD, pp. 1–68. Kluwer Academic, Boston (2000)
6. Carter, J.L.: Dual method for total variation-based image restoration. CAM report 02-13, UCLA, USA; see http://www.math.ucla.edu/applied/cam/index.html. Ph.D. thesis, University of California, LA (2002)
7. Chambolle, A.: An algorithm for total variation minimization and applications. J. Math. Image Vis. **20**, 89–97 (2004)
8. Chambolle, A., Lions, P.L.: Image recovery via total variation minimization and related problems. Numer. Math. **76**(2), 167–188 (1997)
9. Chan, R.H., Wong, C.K.: Sine transform based preconditioners for elliptic problems. Numer. Linear Algebra Appl. **4**, 351–368 (1997)
10. Chan, R.H., Chan, T.F., Wan, W.L.: Multigrid for differential convolution problems arising from image processing. In: Chan R., Chan T.F., Golub G.H. (eds.) Proc. Sci. Comput. Workshop. Springer, see also CAM report 97-20, UCLA, USA (1997)

11. Chan, R.H., Chang, Q.S., Sun, H.W.: Multigrid method for ill-conditioned symmetric Toeplitz systems. SIAM J. Sci. Comput. **19**, 516–529 (1998)
12. Chan, R.H., Ho, C.W., Nikolova, M.: Salt-and-pepper noise removal by median-type noise detectors and detail-preserving regularization. IEEE Trans. Image Process. **14**, 1479–1485 (2005)
13. Chan, T.F., Chen, K.: On a nonlinear multigrid algorithm with primal relaxation for the image total variation minimisation. Numer. Algorithms **41**, 387–411 (2006)
14. Chan, T.F., Mulet, P.: Iterative methods for total variation restoration. CAM report 96-38, UCLA, USA; see http://www.math.ucla.edu/applied/cam/index.html (1996)
15. Chan, T.F., Tai, X.C.: Identification of discontinuous coefficient from elliptic problems using total variation regularization. SIAM J. Sci. Comput. **25**(3), 881–904 (2003)
16. Chan, T.F., Vese, L.: Image segmentation using level sets and the piecewise-constant Mumford-Shah model. UCLA CAM report CAM00-14, USA (2000)
17. Chan, T.F., Golub, G.H., Mulet, P.: A nonlinear primal dual method for total variation based image restoration. SIAM J. Sci. Comput. **20**(6), 1964–1977 (1999)
18. Chang, Q.S., Chern, I.L.: Acceleration methods for total variation-based image denoising. SIAM J. Sci. Comput. **25**, 982–994 (2003)
19. Chen, K.: Matrix Preconditioning Techniques and Applications. Cambridge Monographs on Applied and Computational Mathematics, No. 19. Cambridge University Press, Cambridge (2005)
20. Frohn-Schauf, C., Henn, S., Witsch, K.: Nonlinear multigrid methods for total variation image denoising. Comput. Visual Sci. **7**, 199–206 (2004)
21. Kelley, C.T.: Iterative Methods for Solving Linear and Nonlinear Equations. SIAM Publications (1995)
22. Kenigsberg, A., Kimmel, R., Yavneh, I.: A multigrid approach for fast geodesic active contours. CIS report 2004-06 (2004)
23. Kimmel, R., Yavneh, I.: An algebraic multigrid approach for image analysis. SIAM J. Sci. Comput. **24**(4), 1218–1231 (2003)
24. Li, Y.Y., Santosa, F.: A computational algorithm for minimizing total variation in image restoration. IEEE Trans. Image Process **5**(6), 987–995 (1996)
25. Lysaker, M., Lundervold, A., Tai, X.C.: Noise removal using fourth-order partial differential equation with applications to medical magnetic resonance images in space and time. IEEE Trans. Image Process. **12**(12), 1579–1590 (2003)
26. Malgouyres, F.: Minimizing the total variation under a general convex constraint for image restoration. IEEE Trans. Image Process. **11**(12), 1450–1456 (2002)
27. Nash, S.: A multigrid approach to discretized optimisation problems. J. Optim. Methods Softw. **14**, 99–116 (2000)
28. Osher, S., Fedkiw, R.: Level Set Methods and Dynamic Implicit Surfaces. Springer, New York (2003)
29. Osher, S., Marquina, A.: 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), 387–405 (2000)
30. Osher, S., Sethian, J.: Fronts propagating with curvature dependent speed: algorithms based on Hamilton-Jacobi formulations. J. Comput. Phys. **79**, 12–49 (1988)
31. Rudin, L.I., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. Physica D **60**, 259–268 (1992)
32. Savage, J., Chen, K.: An improved and accelerated nonlinear multigrid method for total-variation denoising. Int. J. Comput. Math. **82**(8), 1001–1015 (2005)
33. Stuben, K.: An introduction to algebraic multigrid. In: Trottenberg, U., Oosterlee, C.W., Schuller, A. (eds.) Multigrid (2000), Appendix A. Also appeared as GMD report 70 from http://www.gmd.de and http://publica.fhg.de/english/index.htm
34. Ta'asan, S.: Multigrid one-shot methods and design strategy. Lecture Note 4 of Von-Karmen Institute Lectures, http://www.math.cmu.edu/~shlomo/VKI-Lectures/lecture4 (1997)
35. Tai, X.C.: Rate of convergence for some constraint decomposition methods for nonlinear variational inequalities. Numer. Math. **93**, 755–786 (2003). Available online at http://www.mi.uib.no/~tai
36. Tai, X.C., Espedal, M.: Rate of convergence of some space decomposition methods for linear and nonlinear problems. SIAM J. Numer. Anal. **35**, 1558–1570 (1998)
37. Tai, X.C., Tseng, P.: Convergence rate analysis of an asynchronous space decomposition method for convex minimization. Math. Comput. **71**, 1105–1135 (2001)
38. Tai, X.C., Xu, J.C.: Global and uniform convergence of subspace correction methods for some convex optimization problems. Math. Comput. **71**, 105–124 (2001)
39. Trottenberg, U., Oosterlee, C.W., Schuller, A.: Multigrid. Academic, London (2001)
40. Vogel, C.R.: A multigrid method for total variation-based image denoising. In: Bowers, K., Lund, J. (eds.) Computation and Control IV. Progress in Systems and Control Theory, vol. 20. Birkhäuser, Boston (1995)

41. Vogel, C.R.: Negative results for multilevel preconditioners in image deblurring. In: Nielson, M. (ed.) Scale-Space Theories in Computer Vision, pp. 292–304. Springer, New York (1999)
42. Vogel, C.R.: Computational Methods for Inverse Problems. SIAM Publications (2002)
43. Vogel, C.R., Oman, M.E.: Iterative methods for total variation denoising. SIAM J. Sci. Stat. Comput. **17**, 227–238 (1996)
44. Vogel, C.R., Oman, M.E.: Fast, robust total variation-based reconstruction of noisy, blurred images. IEEE Trans. Image Process. **7**, 813–824 (1998)
45. Wan, W.L., Chan, T.F., Smith, B.: An energy-minimizing interpolation for robust multigrid methods. SIAM J. Sci. Comput. **21**(4), 1632–1649 (2000)
46. Xu, J.C.: Iteration methods by space decomposition and subspace correction. SIAM Rev. **4**, 581–613 (1992)
47. Yip, A.M., Park, F.: Solution dynamics, causality, and critical behavior of the regularization parameter in total variation denoising problems. CAM report 03-59, UCLA, USA (on-line) (2003)