A NONLINEAR MULTIGRID METHOD FOR CURVATURE EQUATIONS RELATED TO TOTAL VARIATION MINIMIZATION

KE CHEN* AND XUE-CHENG TAI†

Abstract. Image restoration has been an active research topic in the last few years. While much theory as well as many useful models and methods have been developed, there still exists a need to develop fast iterative solvers. This paper proposes to use piecewise linear function spanned subspace correction to design a multilevel method for directly solving the total variation minimisation. Both the theoretical and experimental results are presented.
Keywords: Image restoration, total variation, regularisation, subspace correction, multilevel solvers.
AMS subject class: 68U10, 65F10, 65K10.

1. Introduction. Given a bounded domain $\Omega \subset R^d,\ d = 1, 2, 3, \cdots$, we often need to solve problems which can be written in the following general form:

$$\min_u \left(\int_\Omega |\nabla u| dx + \int_\Omega f(u) dx \right). \tag{1.1}$$

The Euler-Lagrange equation for the above minimization problem is

$$-\nabla \cdot \left(\frac{\nabla u}{|\nabla u|} \right) + f'(u) = 0, \tag{1.2}$$

which is known as a curvature equation [20]. The application of problems (1.1) and (1.2) range from image processing including noise removal [23, 18], segmentation [12, 15], deblurring [4], inverse problems [11] to motion driven by mean curvature [22, 20]. Owing to huge number of applications involved with models (1.1) and (1.2), the demand for fast solvers for these problem is huge. Traditionally, the following methods have been used to solve the equation (1.2):

(i) The fixed point iteration method [1, 32, 35, 36, 33, 34]: Once the coefficients $1/|\nabla \bar{u}|$ are fixed at a previous iteration $\bar{u}$, various iterative solver techniques have been considered [35, 36, 8, 7, 6, 17]. There exist excellent inner solvers but the outer solver can be slow. Further improvements are still useful.

(ii) The explicit time marching scheme [23, 21]: 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 method [9, 10, 4]: 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.

Multigrid method (MGM) is one of the most powerful numerical methods for solving linear and nonlinear elliptic problems [37, 30]. Different efforts have been devoted to developing MGM for the curvature equation (1.2), c.f [33, 6, 24, 2]. However, the success so far is rather limited. The main obstacle for the success of MGM for (1.2) is that the nonlinear diffusion coefficient $|\nabla u|^{-1}$ could be highly oscillatory. It can also

*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]
†Department of Mathematics, University of Bergen, Bergen, Norway. Email: xue-cheng.tai@uib.no. Web: http://www.mi.uib.no/∼tai

be degenerate or has values to be nearly infinity. Recently in [13, 16], the linear algebraic multigrid method [25] was adapted for solving the above PDE in each (outer) step of a fixed iteration while [24] attempted to use the standard multigrid methods with a non-standard and somewhat global smoother. Some other approaches, c.f. [3, 26, 5, 19], have tried to design proper minimization problems over the coarser meshes for the MGM which are in some sense trying to mimic the interpolation and prolongation for a linear MGM. However, the convergence of this kind of method for nonlinear problems is not as good as for linear problems.

In [28, 30, 29, 27], MGM and domain decomposition method (DDM) have been interpreted as space decomposition techniques. Moreover, it is proved that efficiency of DDM and MGM for some nonlinear problems is as good as for linear elliptic problems. The essential idea is to use nonlinear smoothers for the subproblems which respect the minimization problem in the sense that it will guarantee that the energy functional is monotonically decreasing. In this work, we shall use these algorithms for the curvature equation (1.2) and demonstrate numerically that the efficiency of the schemes can be as good as for linear problems.

In Section 2, we introduce the general subspace correction methods for convex functional minimisation. In Section 3, we detail our proposed multilevel algorithm for problem (1.2) and present some preliminary analysis. We present numerical algorithms and experiments in Section 4 for solving both the one-dimensional and two-dimensional image denoising problems. Finally in section 5, we discuss some conclusions and possible future work.

2. Nonlinear space decomposition algorithms. Consider a general minimization problem over a reflexive Banach space V:

$$\min_{v\in V} F(v) \tag{2.1}$$

where F is a strongly convex cost functional. We shall assume that the space V can be decomposed into a sum of smaller subspaces, i.e.

$$V = V_1 + V_2 + \cdots + V_m \ . \tag{2.2}$$

This means that for any $v \in V$, there exists $q_i \in V_i$ such that $v = \sum_{i=1}^m q_i$. Two types of subspace correction methods can be derived based on (2.2), namely the parallel subspace correction (PSC) method and the successive subspace correction (SSC) method, as simple generalisations of the methods for operator equations [37].

Following previous studies [27, 28, 29, 30], the parallel subspace correction method can be described as follows.

ALGORITHM 2.1. *Choose an initial value $u^0 \in V$ and relaxation parameters $\alpha_i > 0$ such that $\sum_{i=1}^m \alpha_i \le 1$.*

1. *For $n \ge 0$, if $u^n \in V$ is defined, then find $p_i^n \in V_i$ in parallel for $i = 1, 2, \cdots, m$ such that*

$$F\left(u^n + p_i^n\right) \le F\left(u^n + q_i\right) \ , \quad \forall q_i \in V_i. \tag{2.3}$$

2. *Set*

$$u^{n+1} = u^n + \sum_{i=1}^m \alpha_i p_i^n \ , \tag{2.4}$$

and go to the next iteration.

The successive subspace correction method can be described as follows:

ALGORITHM 2.2. *Choose an initial value* $u^0 \in V$.

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

$$F\left(u^{n+(i-1)/m} + p_i^n\right) \leq F\left(u^{n+(i-1)/m} + q_i\right) \ , \quad \forall q_i \in V_i. \tag{2.5}$$

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 see that relation between MGM and space decomposition, we shall try to use finite element spaces. Similar explanations can also be given for finite difference approximations. For the 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 $\Omega = \cup_i \tau_j^i$ 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. 2.1 and Fig. 2.2, we plot the basis functions and the refined meshes for a domain in one and two dimensions. 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.

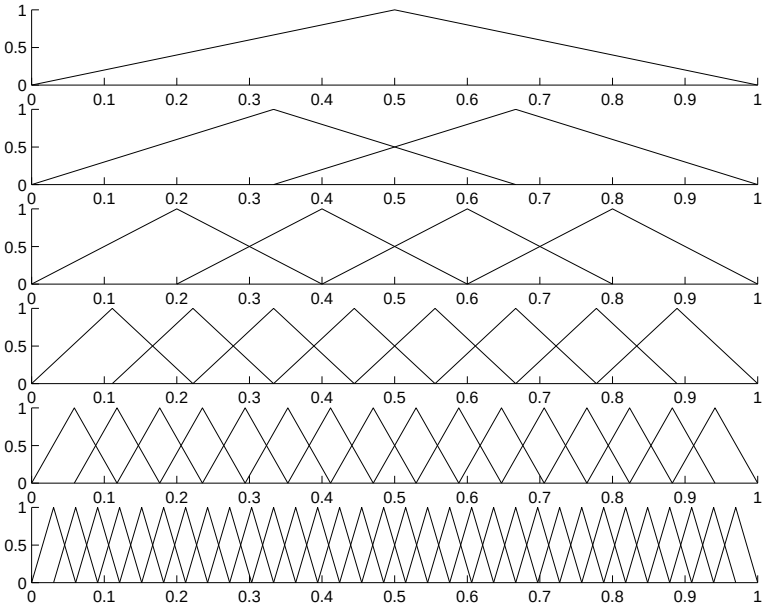


FIG. 2.1. *Basis functions and the mesh for one dimensional multigrids*

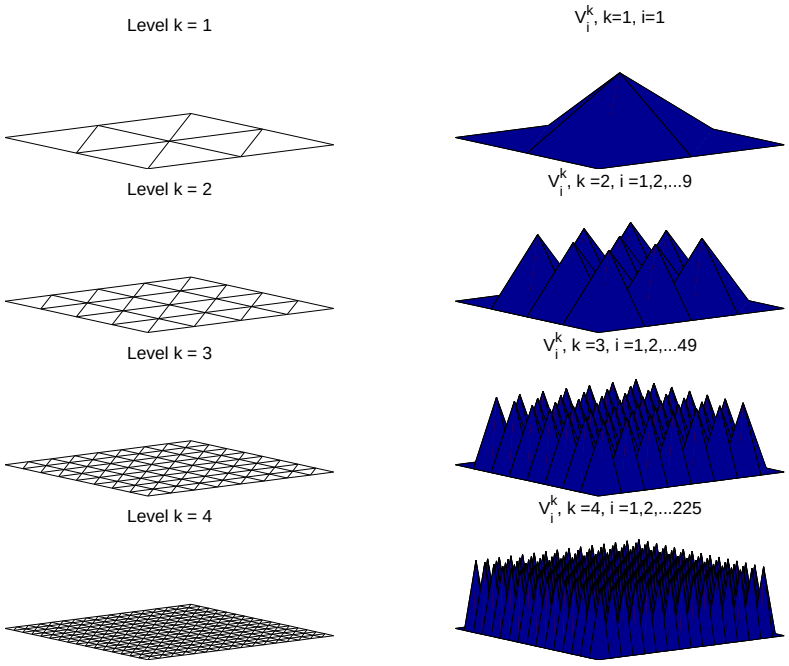


FIG. 2.2. *Basis functions and the mesh for two dimensional multigrids*

Corresponding to each finite element partition $\mathcal{T}_j$, a finite element space $\mathcal{M}_j$ can be defined by

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

Each finite element space $\mathcal{M}_j$ is associated with a nodal basis, denoted by $\{\phi_j^i\}_{i=1}^{n_j}$ satisfying

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

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

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

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

$$V = \sum_{j=1}^{J} \sum_{k=1}^{n_j} V_j^k. \tag{2.6}$$

Each subspace V_j^k is an one dimensional subspace and thus the subproblems (2.5) are easy to solve.

3. Application to noise removal for digital images. For a given noisy image z defined on a domain $\Omega = [0,1] \times [0,1]$, one of the basic models to remove noise from z is to use the ROF model [23]. The ROF model is to take $F(v)$ in (2.1) to be

$$F(v) = \alpha \int_\Omega \sqrt{v_x^2 + v_y^2 + \beta}\; dx + \int_\Omega \|u - z\|^2\; dx. \tag{3.1}$$

The minimizer of (2.1) with F given above is taken as the denoised image. The constant $\alpha > 0$ is chosen according to the noise level and the parameter $\beta > 0$ is chosen to be small. The minimizer u is the solution of

$$-\alpha\nabla\cdot\left(\frac{\nabla u}{\sqrt{|\nabla u|^2+\beta}}\right)+(u-z)=0. \tag{3.2}$$

Instead of solving problem (3.2) directly, we shall proceed to solve its related convex minimisation problem (3.1) by a MGM.

For problem (3.1), we shall explain the details in using Algorithm 2.2 for multigrid decomposition (2.6). Note that all the subspaces in the multi-dimensional decomposition (2.6) are one dimensional. Thus, the subproblems (2.5) are essentially trying to solve the following one dimensional minimization problem:

$$\min_{c\in R} F(w+c\phi_j^k), \tag{3.3}$$

where $w = u^{n+(i-1)/m} \in V$ and ϕ_j^k is the basis function over the jth level at the kth node. As F is strongly convex, x is a minimizer of (3.3) if and only if it satisfies

$$\int_\Omega\left[\alpha\frac{\nabla(w+x\phi_j^k)\cdot\nabla\phi_j^k}{\sqrt{|\nabla(w+x\phi_j^k)|^2+\beta}}+(w+x\phi_j^k-z)\phi_j^k\right]dx=0. \tag{3.4}$$

This one dimensional problem is nonlinear and we may use the fixed point iteration of [1] to solve it, i.e. start with an $x^0 = 0$ and recursively get x^l from

$$\int_\Omega\left[\alpha\frac{\nabla(w+x^{l+1}\phi_j^k)\cdot\nabla\phi_j^k}{\sqrt{|\nabla(w+x^l\phi_j^k)|^2+\beta}}+(w+x^{l+1}\phi_j^k-z)\phi_j^k\right]dx=0. \tag{3.5}$$

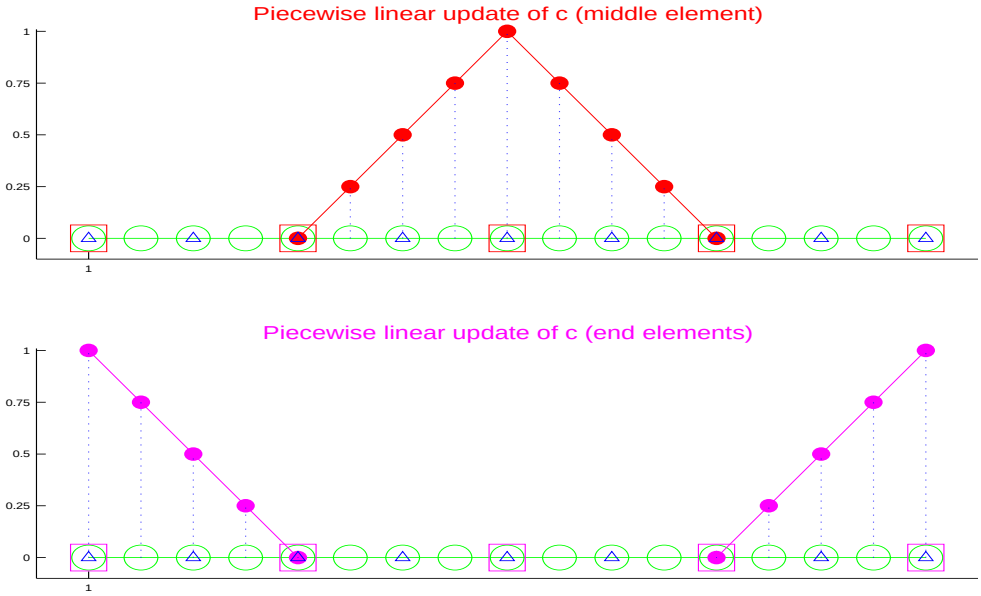
It is easy to see that

$$x^{l+1}=\frac{b_j^k-a_j^k(w)}{a_j^k(\phi_j^k)},\quad b_j^k=\int_\Omega(z-w)\phi_j^k dx \quad \text{and}$$
$$a_j^k(v)=\int_\Omega\left[\alpha\frac{\nabla v\cdot\nabla\phi_j^k}{\sqrt{|\nabla(w+x^l\phi_j^k)|^2+\beta}}+v\phi_j^k\right]dx. \tag{3.6}$$

It is easy to see that $a_j^k(\phi_j^k) > 0$ and thus such an iteration will never break down. As w is a function over the fine mesh, thus we must use the fine mesh elements in doing integrations for getting $a_j^k(w)$ and $a_j^k(\phi_j^k)$. Regarding complexity, we note that the domain integration in (3.5) and (3.6) does not present complications because the basis function ϕ_j^k is only defined locally (as with finite elements). This is addressed in the next section.

4. Numerical algorithms and experiments. To implement Algorithm 2.2 using the above described framework via (3.3-3.6), we now examine the details of solving the discrete problems of these equations for a more efficient implementation. The idea will be applicable to more general setting. Then we shall present some experimental results.

FIG. 4.1. *The one dimensional basis function ϕ_j^k – on the coarse level $j = 2$ and at a middle node $k = 2$ (top plot) and at end nodes (bottom plot).*



4.1. The one dimensional algorithm. Firstly we consider the one dimensional case and note that ϕ_j^k (on level $j = 2$ and at node $k = 2$) may be illustrated by Figure 4.1. We wish to simplify the functional as much as possible by using the compact support of ϕ_j^k. For easier notation, we drop the parameter β (and re-introduce it back later). Then simple manipulations show that (note boundary basis functions require the usual adjustment in indices).

In the discrete setting for one dimensional problems, the cost functional (3.1) with $\beta = 0$ is:

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

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 and we define $I_j^k = \{i |\ x_i \in \Omega_j^k \cap N\}$ and $\bar{I}_j^k = \{i |\ x_i \in \bar{\Omega}_j^k \cap N\}$ with N being the set of the nodal points for the discretization. It is clear that

$$F(u) = \widetilde{F_j^k}(u) + \alpha \sum_{i \in \bar{I}_j^k} |D_x^+ u_i| + \frac{1}{2} \sum_{i \in I_j^k} (u_i - z_i)^2, \tag{4.1}$$

where $\widetilde{F_j^k}$ contains all terms not overlapping with the support of ϕ_j^k. Let $v_i = \phi_j^k(x_i)$.

With the above splitting of the functional

$$F(w+c\phi_j^k) = \widetilde{F_j^k}(w) + \alpha \sum_{i\in \bar{I}_j^k} |D_x^+ w_i - cD_x^+ v_i| + \frac{1}{2}\sum_{i\in I_j^k}(w_i - z_i + cv_i)^2.$$
$$= \widetilde{F_j^k}(w) + \alpha \sum_{i\in \bar{I}_j^k} |D_x^+ w_i - cD_x^+ v_i| \ + \frac{s}{2}(c-z^*)^2, \tag{4.2}$$

where $\bar{z} = z - w$,

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

Therefore, in 1D, solving (3.3) is equivalent to solving

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

and (with β added) the following

$$\min_{c\in R} J(c), \qquad J(c) = \left[\alpha \sum_{i\in \bar{I}_j^k} \sqrt{\left(D_x^+ w_i - cD_x^+ v_i\right)^2 + \beta} + \frac{s}{2}(c-z^*)^2\right]. \tag{4.3}$$

Further, implementing (3.5) and (3.6) for the simplified equation (4.3) leads to the iterations

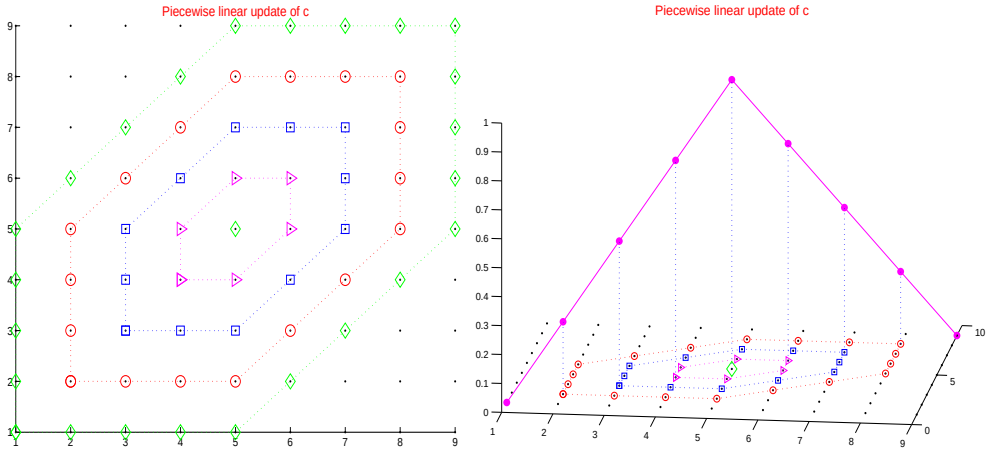
$$\left[\alpha \sum_{i\in \bar{I}_j^k} \frac{|D_x^+ v_i|^2}{\sqrt{\left(D_x^+ w_i - x^l D_x^+ v_i\right)^2 + \beta}} + s\right] x^{l+1} =$$
$$\left[sz^* - \alpha \sum_{i\in I_j^k} \frac{D_x^+ w_i D_x^+ v_i}{\sqrt{\left(D_x^+ w_i - x^l D_x^+ v_i\right)^2 + \beta}}\right], \quad \text{for } l = 1, 2, \cdots. \tag{4.4}$$

4.2. The two dimensional algorithm. Secondly we can apply the same argument of simplification to the 2D case, where we note that a 2D basis function ϕ_j^k (similar to Figure 4.1) may be illustrated by Figure 4.2. That is, the terms in the functional $F(w + c\phi_j^k)$, $c \in R$, from (3.3) may again be grouped and simplified according to the compact support of ϕ_j^k. Similar to the 1D case in (4.2), the values of the 2D basis function may be denoted by matrix v, which takes the values

$$v = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{4} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{4} & 0 & 0 & 0 \\ 0 & \frac{1}{4} & \frac{1}{2} & \frac{3}{4} & \frac{3}{4} & \frac{1}{2} & \frac{1}{4} & 0 & 0 \\ 0 & \frac{1}{4} & \frac{1}{2} & \frac{3}{4} & 1 & \frac{3}{4} & \frac{1}{2} & \frac{1}{4} & 0 \\ 0 & 0 & \frac{1}{4} & \frac{1}{2} & \frac{3}{4} & \frac{3}{4} & \frac{1}{2} & \frac{1}{4} & 0 \\ 0 & 0 & 0 & \frac{1}{4} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{4} & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{4.5}$$

for the example of $j = 3$ and $b = b_j = 4$ (as in Figure 4.2) and when we zoom in the neighbourhood of index k (as v is actually a global quantity with a compact support).

FIG. 4.2. *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 (4.5), are shown.)*



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. In the discretized setting, we have

$$F(u) = \alpha \sum_{k_1=1}^{n} \sum_{k_2=1}^{m} \sqrt{(D_x^+ u_{k_1,k_2})^2 + (D_y^+ u_{k_1,k_2})^2} + \frac{1}{2} \sum_{k_1=1}^{n} \sum_{k_2=1}^{m} (u_{k_1,k_2} - z_{k_1,k_2})^2 \quad (4.6)$$

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 $F(w + c\phi_j^k)$ as follows:

$$\begin{aligned} F(w + c\phi_j^k) &= F(w + cv) = \widetilde{F_j^k}(w) + \\ &\alpha \sum_{(k_1,k_2)\in \bar{I}_j^k} \sqrt{(D_x^+ w_{k_1,k_2} + cD_x^+ v_{k_1,k_2})^2 + (D_y^+ w_{k_1,k_2} + cD_y^+ v_{k_1,k_2})^2 + \beta} \\ &\qquad + \frac{1}{2} \sum_{(k_1,k_2)\in I_j^k} (\bar{z}_{k_1,k_2} - cv_{k_1,k_2})^2, \\ &= \overline{F_j^k}(w, \bar{z}, v) + \alpha \sum_{(k_1,k_2)\in \bar{I}_j^k} T_{k_1,k_2}(c) + \frac{s}{2}(c - z^*)^2, \end{aligned} \quad (4.7)$$

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

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

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

Omitting the details, we find that the updating of (3.6) or (3.5) in the discretized setting for 2D problem is:

$$\left[\alpha \sum_{(k_1,k_2)\in \bar{I}_j^k} \frac{|D_x^+ v_{k_1,k_2}|^2 + |D_y^+ v_{k_1,k_2}|^2}{T_{k_1,k_2}(x^l)} + s\right] x^{l+1} =$$
$$\left[sz^* - \alpha \sum_{(k_1,k_2)\in \bar{I}_j^k} \frac{D_x^+ w_{k_1,k_2} D_x^+ v_{k_1,k_2} + D_y^+ w_{k_1,k_2} D_y^+ v_{k_1,k_2}}{T_{k_1,k_2}(x^l)}\right], \quad \text{for } l = 1, 2, \cdots . \tag{4.8}$$

The iteration for (3.5) is stopped when $|a_j^k(w + x^l\phi_j^k) - b_j^k| \le \tau_{inner}$. Numerical experiments will show that the convergence rate is nearly independent of τ_{inner}. Normally, just carrying out one or two iterations for (3.4) is sufficient to get rather good convergence rate. For linear problems, the cost per iteration for the multigrid iteration can be reduced to $O(DOF)$ where DOF is the total number of degree of freedom. For the nonlinear problem here, the cost per iteration is $O(DOF \log(DOF))$.

Finally we present some numerical experiments. The above proposed algorithm (Algorithm 2.2 with (3.3) as a core step) has not been applied to the image total variation minimisation before. Here we shall first test its effectiveness in solving the image total variation minimisation for some image denoising problems. Then we experiment on the dependence of the convergence of the proposed multigrid algorithm on the image problem sizes. Finally we experiment on the influence of the inner Picard fixed point iterations (3.4) 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.

Test problems and results. We shall consider 4 one-dimensional problems as shown in Figure 4.3 and another 4 two-dimensional problems as shown in Figure 4.5. 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 $\|u^{k+1} - u^k\|_2/\|u^{k+1}\|_2 < \tau_{outer}$ for a prescribed tolerance τ_{outer}. Then $k+1$ will be the number of outer iteration steps. There is another prescribed tolerance τ_{inner} which is to control how accuracy the iterations should be in the solution of the local minimisation (3.3). Here we take $\tau_{inner} = \tau_{outer} = 10^{-3}$ and $\beta = 10^{-4}$ for the regularising parameter. The processed results by our algorithm is shown in Figure 4.4 (for $N = 4097$) and Figure 4.6 (for $N \times N = 257 \times 257$) respectively, where the symbol $\square$ refers to our algorithm while the symbol $\times$ the method of [9]. Clearly one observes that our method converges quite quickly and gives a result which is not distinguishable from the result of [9].

4.3. 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 and $n \times n$ in 2D). In Table 4.1, we fix both tolerances $\tau_{inner}, \tau_{outer}$ 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 approaches a constant as n increases.

4.4. 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 4.2 shows the results obtained for the

FIG. 4.3. *The 1D test examples*

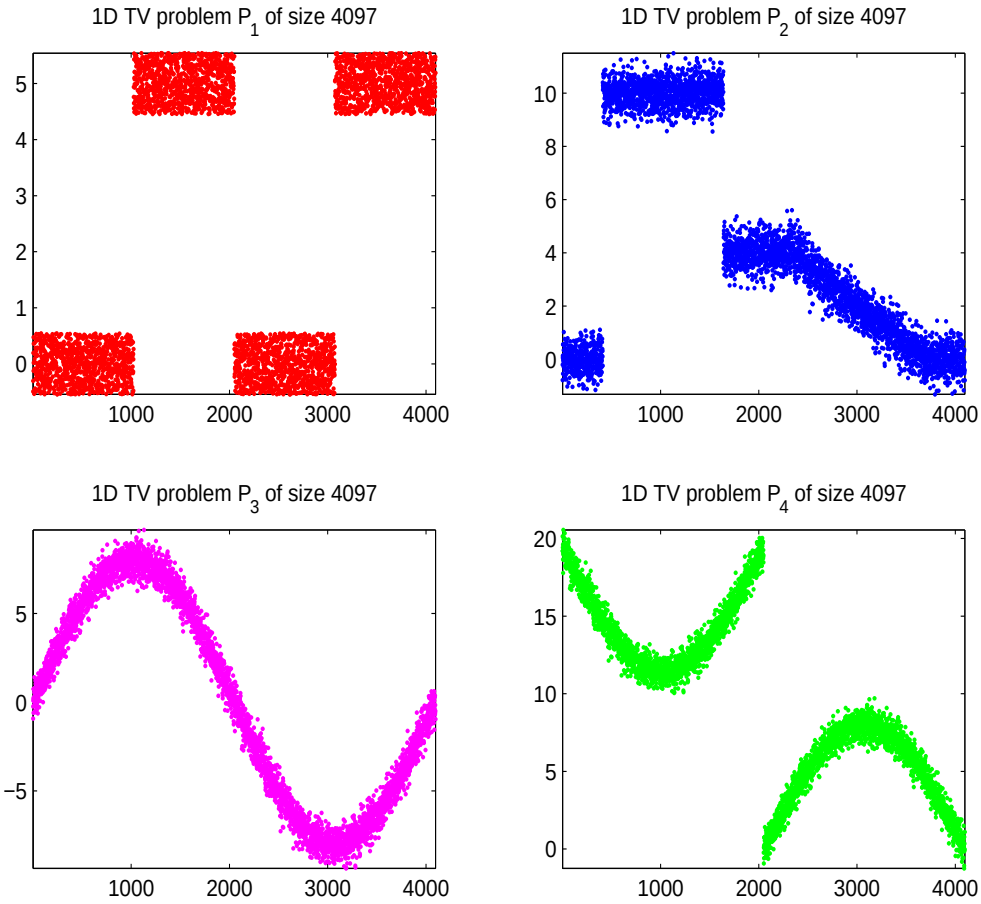


FIG. 4.4. *The 1D processed results:* □ *- the new multilevel algorithm and* × *- the primal-dual method [9].*

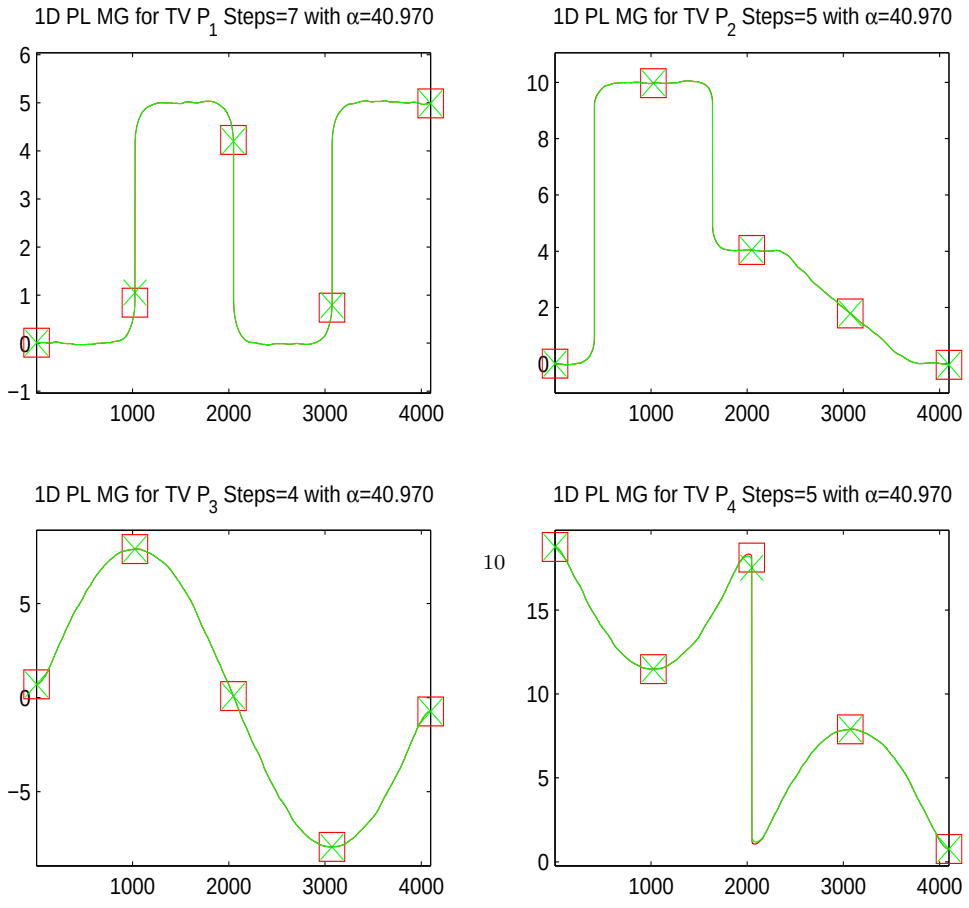


FIG. 4.5. *The 2D test examples*

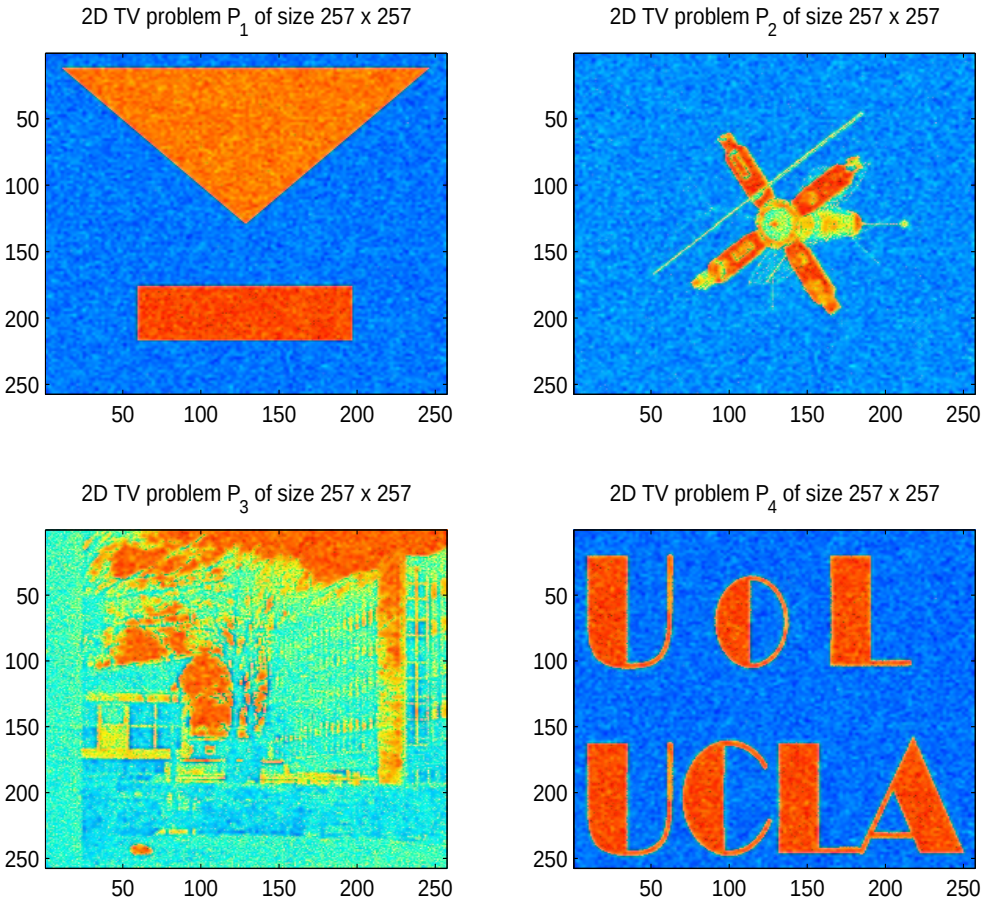


FIG. 4.6. *The 2D processed results:* □ *- the new multilevel algorithm and* × *- the primal-dual method [9].*

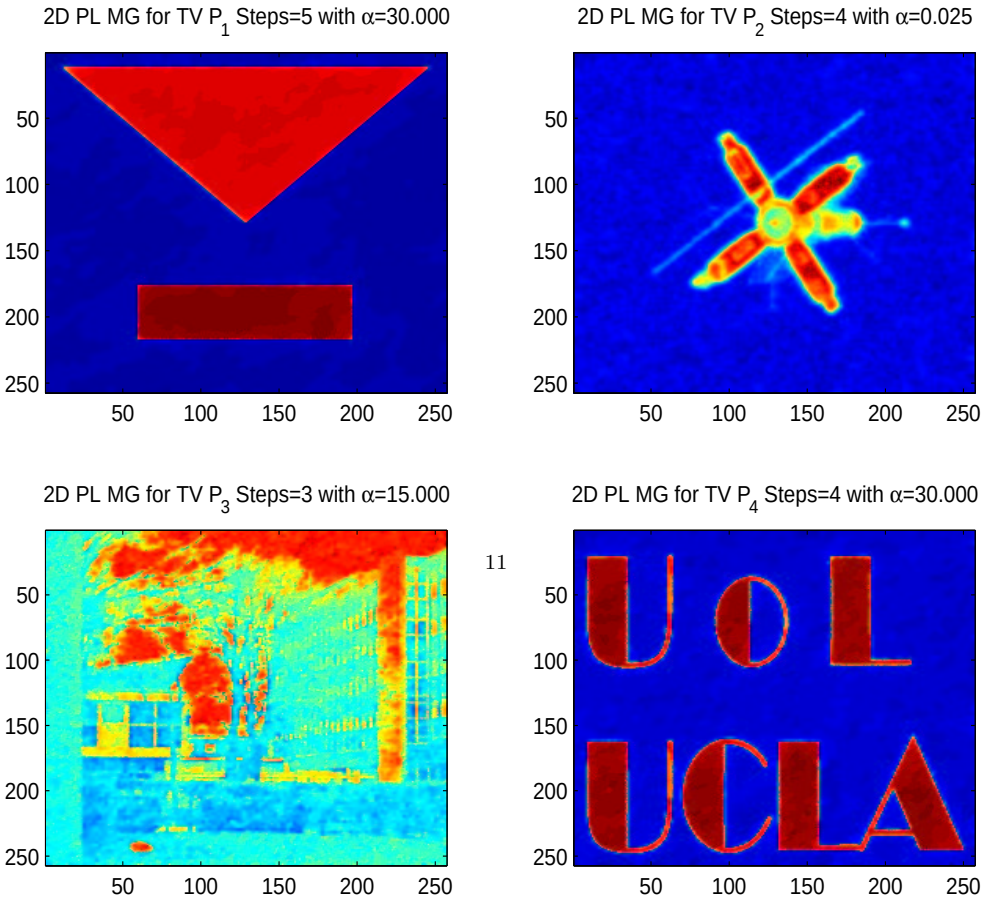


TABLE 4.1

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 steps'. Here $\tau = 10^{-3}$, $\beta = 10^{-4}$. Clearly there is no strong dependence. Here the Problem numbers refer to Figure 4.3 for 1D and Figure 4.5 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

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 largest tolerance $\tau_{inner} = 10$, 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 [14] and adopted in the algorithm of [9]. It is possible to work out an appropriate formula for τ_{inner}.

5. Conclusions. This paper has considered a nonlinear multigrid method for solving curvature equations related to total variation minimization. Numerical tests show that the convergence for the MG algorithm is mesh independent as long as α is

TABLE 4.2

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 steps'. 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 Figure 4.3 for 1D and Figure 4.5 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

not chosen to be very large, i.e. $\alpha \in (0, 1]$. The parameter β is used to control the width of the jumps. The smallest width is over one mesh element. As long as β is chosen in this range, the algorithm offers mesh independent convergence which is also not very sensitive to the values of β. However, we can take $\beta \approx 0$ for experimental purposes. For very small β, overflow may happen during the iterations and we will then have problems to have our algorithms to converge. It is known that the algorithm of [9] is rather robust with respect to α and β. For most of the experiments we have done, we need less than 20 iterations to get a results which is nearly undistinguishable from the results of [9].

Even mesh independent convergence is observed in our numerical simulations. A full theoretical justification is still missing for small β (including $\beta = 0$). In order to use the existing theory to justify this convergence behaviour, we need to verify

that the functional F and the decomposed spaces satisfy some theoretical conditions (relating to differentiability), see [27, p.212] and [30, p.121]. Up to now, it is still an open problem to find a proper norm for F to satisfy the needed conditions. Although the standard total variation norm does not satisfy the required conditions, as the problem (1.2) is a nonlinear diffusion process and the subproblem iteration procedure is reducing the cost functional all the time, thus it respects the nonlinear diffusion behavior of the problem and corrects the errors up when going to finer and finer meshes.

REFERENCES

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

[2] S. T. Acton. Multigrid anisotropic diffusion. *IEEE Trans. Imag. Proc.*, 3 (3):280–291, 1998.

[3] E. Arian and S. Ta'asan. Multigrid one-shot methods for optimal control problems. *ICASE technical report No. 94-52, USA*, 1994.

[4] P. Blomgren, T. F. Chan, P. Mulet, L. Vese, and W. L. Wan. Variational PDE models and methods for image processing. In *Research Notes in Mathematics*, volume 420, pages 43–67. Chapman & Hall/CRC, 2000.

[5] A. Brandt. Multigrid solvers and multilevel optimization strategies. In J. Cong and J. R. Shinnerl, editors, *Multiscale Optimization and VLSI/CAD*, pages 1–68. Kluwer Academic (Boston), 2000.

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

[7] R. H. Chan, Q. S. Chang, and H. W. Sun. Multigrid method for ill-conditioned symmetric Toeplitz systems. *SIAM J. Sci. Comput.*, 19:516–529, 1998.

[8] R. H. Chan and C. K. Wong. Sine transform based preconditioners for elliptic problems. *Numer. Linear Algebra Applic.*, 4:351–368, 1997.

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

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

[11] T. F. Chan and X. C. Tai. Identification of discontinuous coefficient from elliptic problems using total varaition regularization. *SIAM J. Sci. Comput.*, 25 (3):881–904, 2003.

[12] T. F. Chan and L. Vese. Image segmentation using level sets and the piecewise-constant mumford-shah model. *UCLA CAM report CAM00-14, USA*, 2000.

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

[14] C. T. Kelley. *Iterative Methods for Solving Linear and Nonlinear Equations*. SIAM publications, USA, 1995.

[15] A. Kenigsberg, R. Kimmel, and I. Yavneh. A multigrid approach for fast geodesic active contours. *CIS Report 2004-06*, 2004.

[16] R. Kimmel and I. Yavneh. An algebraic multigrid approach for image analysis. *SIAM J. Sci. Comput.*, 24(4):1218–1231, 2003.

[17] Y. Y. Li and F. Santosa. A computational algorithm for minimizing total variation in image restoration. *IEEE Trans. Image Proc.*, to appear., 1996.

[18] M. Lysaker, A. Lundervold, and X. C. Tai. Noise removal using fourth-order partial differential equation with applications to medical magnetic resonance images in space and time. *IEEE Transactions on Image Processing*, 12 (12):1579–1590, 2003.

[19] S. Nash. A multigrid approach to discretized optimisation problems. *J. Opt. Methods Softw.*, 14:99–116, 2000.

[20] S. Osher and R. Fedkiw. *Level Set Methods and Dynamic Implicit Surfaces*. Springer, 2003.

[21] S. Osher and A. Marquina. 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.

[22] S. Osher and J. Sethian. Fronts propagating with curvature dependent speed: algorithms based on Hamilton-Jacobi formulations. *J. Comp. Phys.*, 79:12–49, 1988.

[23] L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D*, 60:259–268, 1992.

[24] J. Savage and K. Chen. An improved and accelerated nonlinear multigrid method for total-variation denoising. *submitted*, 2004.

[25] K. Stuben. *An introduction to algebraic multigrid, in Appendix A of [31].* Also appeared as GMD report 70 from `http://www.gmd.de` and `http://publica.fhg.de/english/index.htm`, 2000.

[26] S. Ta'asan. *Multigrid one-shot methods and design startegy.* Lecture note 4 of Von-Karmen Institute Lectures, `http://www.math.cmu.edu/~shlomo/VKI-Lectures/lecture4`, 1997.

[27] X. C. Tai. Rate of convergence for some constraint decomposition methods for nonlinear variational inequalities. *Numer. Math.*, 93:755–786, 2000. Available online at http://www.mi.uib.no/~tai.

[28] X. C. Tai and M. Espedal. Rate of convergence of some space decomposition methods for linear and nonlinear problems. *SIAM. J. Numer. Anal.*, 35:1558–1570, 1998.

[29] X. C. Tai and P. Tseng. Convergence rate analysis of an asynchronous space decompostion method for convex minimization. *Math. Comp.*, 71:1105–1135, 2001.

[30] X. C. Tai and J. C. Xu. Global and uniform convergence of subspace correction methods for some convex optimization problems. *Math. Comp.*, 71:105–124, 2001.

[31] U. Trottenberg, C. W. Oosterlee, and A. Schuller. *Multigrid.* Academic Press, London, UK, 2000.

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

[33] C. R. Vogel. Negative results for multilevel preconditioners in image deblurring. In M. Nielson et al, editor, *Scale-Space Theories In Computer Vision*, pages 292–304. Springer-Verlag, 1999.

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

[35] C. R. Vogel and M. E. Oman. Iterative methods for total variation denoising. *SIAM J. Sci. Statist. Comput.*, 17:227–238, 1996.

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

[37] J. C. Xu. Iteration methods by space decomposition and subspace correction. *SIAM Rev.*, 4:581–613, 1992.