ACCELERATION METHODS FOR TOTAL VARIATION-BASED IMAGE DENOISING

QIANSHUN CHANG * AND I-LIANG CHERN †

Abstract. In this paper, we apply a fixed point method to solve the total variation-based image denoising problem. An algebraic multigrid method is used to solve the corresponding linear equations. Krylov subspace acceleration is adopted to improve convergence in the fixed point iteration. A good initial guess for this outer iteration at finest grid is obtained by combining fixed point iteration and geometric multigrid interpolation successively from the coarsest grid to the finest grid. Numerical experiments demonstrate that this method is efficient and robust even for images with large noise-to-signal ratios.

Key words. Image denoising, total variation, fixed point method, algebraic multigrid method, Krylov acceleration

AMS subject classifications. 65F30

1. Introduction.

The image restoration problem is to recover a "true" image u from an observed image z. The latter is usually noisy and blurred, and is modeled by $z = Ku + n$. Here, K is a known linear blur operator, and n is a Gaussian white noise. When K is the identity, the observed images are not blurred. and the corresponding problem is called a denoising problem. In recent years, one main method for noise removal and deblurring is the total variation based restoration method, proposed by L. Rudin, S. Osher and E. Fatemi [16]. In this method, the total variation of u is used as a regularization penalty functional for a corresponding minimization problem (see (1.1) below). The main advantage of this method is that it can maintain sharpness of edges of images. The drawback is that the corresponding partial differential equation (PDE) is harder to solve because the degeneracy of the diffusion coefficients on edges. This paper is devoted to an efficient algorithm for solving this nonlinear PDE.

Using the Tikhonov penalty method and a diffusion regularization, the total variation based restoration method can be formulated as an unconstrained minimization problem:

$$\min_u \left(\alpha \int_\Omega \sqrt{| \nabla u|^2 + \beta}\ dxdy + \frac{1}{2} \|Ku - z\|^2_{L^2} \right). \tag{1.1}$$

Here, $\alpha > 0$ is the penalty parameter which controls the trade-off between the smoothness of u and the goodness of fit-to-the-data. The parameter $\beta > 0$ is a regularization parameter and is usually small. The functional in (1.1) is strictly convex. Its global minimizer is unique. The well-posedness of problem (1.1) with $\beta \to 0+$ has been discussed in [1].

The corresponding Euler-Lagrange equation for (1.1) is

$$-\alpha \nabla \cdot (\frac{\nabla u}{\sqrt{|\nabla u|^2 + \beta}}) + K^*(Ku - z) = 0, \tag{1.2}$$

where K^* is the adjoint operator of K with respect to the L_2 inner product. Numerically, solving equation (1.2) is not an easy task for its nonlinearity and singularity in

[*] Institute of Applied Mathematics, Academy of Mathematics and Systems Sciences, Chinese academy of Sciences, Beijing, China (cqs@amath8.amt.ac.cn).

[†] Department of Mathematics, National Taiwan University, Taipei, Taiwan (chern@math.ntu.edu.tw), supported by grant NSC90-2115-M-002-020.

the diffusion term. Many methods have been proposed. The time marching scheme to reach the steady state of the corresponding parabolic equation of (1.2) was suggested in [3, 16]. An affine scaling algorithm was proposed in [12]. Vogel and Oman [20] applied a fixed point method to solve equation (1.2). Global convergence of fixed point method in finite space setting was shown in [10]. Newton's method with a continuation procedure on the regularization parameter β was used in [6]. T. Chan, G. Golub, and P. Mulet [7] proposed a nonlinear primal-dual method. A multigrid method was proposed to solve the linearization part of equation (1.2) in [19, 14, 18]. However, the convergence rate there was slow.

In this paper, we focus on denoising problems. We suggest to use the fixed point method [20] with Krylov's acceleration procedure instead of Newton's method, despite the latter is of quadratic convergence. From our experience, for practical images with reasonable accuracy, the number of iterations needed in Newton's method is about the same as that of a fixed point method with algebraic multigrid acceleration. The reason is that, the corresponding linear system in Newton's method is much harder to solve, because its diffusion coefficient (which is a matrix) is always degenerate in certain direction (the normal direction of the level set of the underlying function u), while the diffusion coefficient of (1.2) is a scalar, thus no preferential direction of degeneracy. Although this diffusion coefficient may vary dramatically, it can still be manageable. An algebraic multigrid (AMG) method is suitable to solve the corresponding linear system. We will show that the number of inner iterations needed is much less than that of a preconditioned conjugate gradient method used in Newton's method.

Multigrid methods have been successful for solving elliptic or parabolic problems numerically. The method is nearly optimal in the sense that the computational work required to achieve a fixed accuracy is proportional to the number of discrete unknowns [13]. The algebraic multigrid (AMG) method is designed to utilize the principle of the geometrically oriented multigrid (GMG) method to obtain a fast and automatic solution procedure for linear algebraic systems of equations (see [8, 9, 17]). This method is particularly suitable to our problem, where the coefficients may vary dramatically. AMG method has been adopted as a preconditioner for image restoration problem [18]. In this paper, we adopt an improved version of ordinary AMG method [8]. The number of inner iterations needed is just one per each outer iteration. The convergent factor about 0.05. This is much better than Newton's method with PCG, where the numbers of inner iteration are between 4 to 7 [7].

By noticing the convexity of the functional (1.1), or equivalently, the monotonicity of equation (1.2), we adopt the Krylov subspace method [5, 15] to accelerate the outer iteration in our fixed point method. The improvement is about double in our numerical experiment.

The fixed point method with small β converges slowly [10]. A nice initial guess in a fixed point method can allow us to choose small β and meantimes to reduce the number of iterations. No continuation procedure in β is needed as that in [6]. To produce such a nice initial guess at finest grid, we first solve (1.2) at coarsest grid (by using the fixed point method with a direct solver), then we interpolate to the next fine grid level, then solve equation (1.2) again at that level with algebraic multigrid solver, and so on, until we reach the finest grid. In this grid continuation procedure, a natural diffusion regularization is introduced implicitly because there is no diffusion degeneracy problem at coarse grid at all. In our experiment, the improvement from this grid continuation procedure is about 40%. But it can allow us to choose $\beta = 10^{-32}$, the smallest positive machine number.

This accelerated fixed point method with algebraic multigrid procedure and a nice initialization procedure produces a valuable computational method for image denoising. Computational experiments demonstrate that the method is efficient and robust even for images with large noise-to-signal ratios.

The paper is organized as follows. Section 2 is the finite difference scheme and the fixed point algorithm. In section 3, we introduce the algebraic multigrid algorithm to solve the corresponding linear systems. The combination of Krylov subspace acceleration and the fixed point iteration is given in section 4. And section 5 is the initialization procedure to produce a nice initial data for fixed point iteration. Finally, numerical results and discussion are given in section 6.

2. Difference Scheme and Fixed Point Method. Let us consider the Euler-Lagrange equation

$$-\alpha\nabla\nabla\cdot(\frac{\nabla u}{\sqrt{|\nabla u|^2+\beta}})+K^*(Ku-z)=0 \text{ in } \ \Omega=(0,1)\times(0,1), \tag{2.1}$$

with zero Neumann (no flux) boundary condition. We partition the domain $(0,1)\times(0,1)$ into $L\times L$ uniform cells. Denote $1/L$ by h. The cell centers are $(x_l,y_k)=((l-1/2)h,(k-1/2)h)\,,l,k=1,\cdots,L$. The value $u(x_l,y_k)$ is approximated by $u_{l,k}$. Following [20], we discretize (1.2) by a standard five-point finite difference scheme:

$$\begin{array}{l} -\frac{1}{h^2}\left[(D_{l+1/2,k}+D_{l-1/2,k}+D_{l,k+1/2}+D_{l,k-1/2})\,u_{l,k}\right. \\ \quad\left. -D_{l+1/2,k}u_{l+1,k}-D_{l-1/2,k}u_{l-1,k}-D_{l,k+1/2}u_{l,k+1}-D_{l,k-1/2}u_{l,k-1}\right] \\ \quad +\left(K^*(KU-Z)\right)_{l,k}=0, \quad l,k=1,\cdots,L, \end{array} \tag{2.2}$$

where

$$D_{l+1/2,k}=\frac{\alpha}{\sqrt{|(u_{l+1,k}-u_{l,k})/h|^2+\beta}}, \tag{2.3}$$

and $U=(u_{1,1},u_{1,2},\cdots,u_{1,L},u_{2,1},\cdots,u_{2,L},\cdots,u_{L,L})$, $Z=(z_{1,1},z_{1,2},\cdots,z_{1,L},z_{2,1},\cdots,z_{L,L})$. The discrete Neumann boundary conditions is

$$u_{0,k}=u_{1,k}, \quad u_{L+1,k}=u_{L,k}, \quad u_{l,0}=u_{l,1}, \quad u_{l,L+1}=u_{l,L}. \tag{2.4}$$

We abbreviate (2.2) by

$$A(U)U+K^*(KU-Z)=0. \tag{2.5}$$

Following [20], we use the following fixed point method to solve the above finite difference equation.

$$A(U^s)U^{s+1}+K^*(KU^{s+1}-Z)=0. \tag{2.6}$$

In order to solve system (2.6) for u^{s+1} efficiently, Vogel [19] applied the geometric multigrid (GMG) method. Unfortunately, he found that the GMG converges slowly, because the diffusion coefficients $1/\sqrt{|\nabla u^s|^2+\beta}$ varies too much. In this paper we use the algebraic multigrid (AMG) method instead, where the information of these large variation of coefficients are built in the interpolation operator and coarse grid equation. Near edges (where the coefficients $1/\sqrt{|\nabla u^s|^2+\beta}$ is very small), it maintains more grid points than the GMG method in the coarse grid equation. Hence, the coarse grid equation is more accurate and the convergence rate is improved.

3. Basic AMG Algorithm. Now, we describe our version of the AMG algorithm [8, 17] briefly. We consider the following $n \times n$ system of linear equations

$$AU = F. \tag{3.1}$$

An AMG method breaks this equation into a sequence of smaller and smaller equations: $A^m U^m = F^m, m = 1, \cdots, M$, where $A^m = (a_{i,j}^m)_{n_m \times n_m}$, $U^m = (u_1^m, u_2^m, \cdots, u_{n_m}^m)^T$, and $F^m = (f_1^m, f_2^m, \cdots, {}_{n_m}^m)^T$, with $n = n_1 > n_2 > \cdots > n_M$, $A^1 = A, U^1 = U, F^1 = F$. These equations formally play the same role as the coarse grid equations in the GMG method.

In a standard multigrid process, one needs to define the coarse grids, the interpolation operator I_{m+1}^m, the restriction operator I_m^{m+1}, and the coarse grid operator A^{m+1}. With these, at each level, a smoothing process, say Gauss-Seidel, is applied to the equation $A^m U^m = F^m$ to find an approximate solution $\widehat{U}^m$. The high frequency errors of the residual $r^m := F^m - A^m \widehat{U}^m$ are usually reduced in this smoothing process. The correction for low frequency errors is approximated by the following procedure. First, the correction equation $A^m e^m = r^m$ is restricted to the next coarser grid by the restriction operator. The resulting equation is solved to obtain the coarse grid correction e^{m+1}. This correction e^{m+1} is then interpolated back to level m by the interpolation operator to obtain approximate solution e^m.

We shall adopt Galerkin type algorithm, where $I_m^{m+1} = (I_{m+1}^m)^T$ and $A^{m+1} = I_m^{m+1} A^m I_{m+1}^m$. Thus, we will only need to define the coarse grids and interpolation operators. We follow the approach in [8, 17] to define the grid Ω^m and its coarse grid C^m. The grid Ω^m is regarded as the indices $\{1, \cdots, n_m\}$ of the unknowns e_j^m, $1 \leq j \leq n_m$. The coarse grid C^m is a subset of Ω^m. The grid Ω^{m+1} is nothing but a re-indexing of C^m. We denote $\Omega^m - C^m$ by F^m, the fine grid. Criteria to determine C^m will be discussed later.

The interpolation operator I_{m+1}^m maps data on Ω^{m+1} to data on Ω^m. Namely, for $i \in C^m$, the datum e_i^m is taken to be the datum on the corresponding index on Ω^{m+1}; while for $i \in F^m$, e_i^m is interpolated from data on C^m. Roughly speaking, this interpolation formula is derived so that the i^{th} equation

$$a_{i,i}^m e_i^m + \sum_{j \in N_i^m} a_{i,j}^m e_j^m = r_i^m \approx 0 \tag{3.2}$$

is almost satisfied. Here, $N_i^m = \{j \in \Omega^m \mid a_{i,j}^m \neq 0,\ j \neq i, \}$, which can be thought as the neighbors of i.

In order to solve (3.2) approximately, we classify the neighbors of the point i into two classes. A point $j \in N_i^m$ is said to be strongly connected to i if

$$\mid a_{i,j}^m \mid \geq \theta \cdot \max_{k \neq i} \mid a_{i,k}^m \mid$$

for some fixed $0 < \theta \leq 1$, and weakly connected if otherwise. We denote the collection of these neighboring points by S_i^m (strong) and W_i^m (weak), respectively. We also denote $C^m \bigcap S_i^m$ by C_i^m. Our goal is to derive an interpolation formula

$$e_i^m = \sum_{j \in C_i^m} \omega_{i,j} e_j^m, \text{ for } i \in F^m$$

so that the i^{th} correction equation is almost satisfied:

$$a_{i,i}^m e_i^m + \sum_{j \in N_i^m} a_{i,j}^m e_j^m = 0. \tag{3.3}$$

Noting that $N_i^m = S_i^m \cup W_i^m = (S_i^m \cap C^m) \cup (S_i^m \cap F^m) \cup W_i^m$, the issue here is how to approximate e_j^m with $j \in S_i^m \cap F^m$ or $j \in W_i^m$ in terms of e_i or e_k^m with $k \in C_i^m$. Before going to the discussion of this issue, let us describe how to choose the coarse grid C^m for a moment.

The coarse grid C^m is chosen such that the following criteria are satisfied:

(C1) For each point $i \in F^m$, every point $j \in S_i^m$ is either in C_i^m or strongly connected to at least one point in C_i^m (i.e. $S_j^m \cap C_i^m \neq \phi$).

(C2) C^m should be the maximal subset of all points with the property that any two points in C^m are not strongly connected to each other.

Condition (C1) ensures that for $i \in F^m$, e_i^m can be constructed from the values e_k^m with $k \in C_i^m$ with certain accuracy. Condition (C2) means that C^m is chosen as smaller as possible to gain efficiency. In general, it is difficult to construct C^m to satisfy (C2) strictly. Ruge and Stüben [17] provided an $O(n_m)$ algorithm to construct the coarse grid C^m which is small enough and leads to linear computational complexity of the overall algorithm practically.

Let us go back to the issue: how to approximate e_j^m with $j \in S_i^m \cap F^m$ or $j \in W_i^m$ in terms of e_i^m or e_k^m with $k \in C_i^m$? For $j \in W_i^m$, we may simply approximate e_j^m by

$$e_j^m = e_i^m, \tag{3.4}$$

based on the smoothness of e^m which we do expect. For $j \in S_i^m \cap F^m$, we look into the j^{th} equation:

$$a_{j,j}^m e_j^m + \sum_{k \in C_i^m \cap N_j^m} a_{j,k}^m e_k^m + \cdots \approx 0.$$

The part "$\cdots$" is secondary error and thus negligible. A natural approximation of e_j^m is the following average formula:

$$e_j^m = \sum_{k \in C_i^m \cap N_j^m} g_{j,k}^m e_k^m, \quad g_{j,k}^m = \frac{|a_{j,k}^m|}{\sum_{\ell \in C_i^m \cap N_j^m} |a_{j,\ell}^m|} \tag{3.5}$$

The condition (C1) (i.e. $C_i^m \cap S_j^m \neq \phi$) guarantees that $\sum_{k \in C_i^m} |a_{j,k}^m|$ is not too small.

The above interpolation formula was given by J. Ruge and K. St$\ddot{u}$ben [17]. An improved interpolation formula using some geometric assumptions was proposed in [8]. It further uses interpolation or extrapolation of formulae (3.4) and (3.5), depending on the "relative locations" of points j, i and $\{k | k \in C_i^m \cap S_j^m\}$. These "geometric" assumptions are as below.

(G1) Elements in N_i^m are the neighbors of a point i in Ω^m. Further, the larger the quantity $|a_{i,j}^m|$ is, the closer the point j is to the point i.

(G2) If $a_{i,j}^m < 0$ or $|a_{i,j}^m|$ is small, we say that the error between i and j is geometrically smooth. Otherwise, we call it geometrically oscillating. Here, we have normalized $a_{i,i} > 0$.

Roughly speaking,"geometrically," the average location of points in $C^m \cap S_i^m \cap S_j^m$ is somewhere between i and j. Therefore the error e_j^m can be approximated more accurately by interpolation or extrapolation using e_i and $\sum_{k \in C_i^m \cap S_j^m} g_{j,k} e_k^m$. More precisely, let us define

$$\zeta_{i,j}^m = \frac{-\sum_{k \in C_i^m \cap N_j^m} a_{j,k}^m}{\sum_{k \in C_i^m \cap N_j^m} \mid a_{j,k}^m \mid}, \qquad \eta_{i,j}^m = \frac{\mid a_{i,j}^m \mid}{\frac{1}{|C_i^m \cap N_j^m|} \sum_{k \in C_i^m \cap N_j^m} \mid a_{j,k}^m \mid}.$$

The quantity $\zeta_{i,j}^m$ indicates whether there is a large negative entry $a_{j,k}^m$ for $k \in C_i^m \cap N_j^m$. When $\zeta \geq 1/2$ and $a_{i,j}^m < 0$, it can be shown that the errors between the point i and the point j are geometrically smooth. The quantity $\eta_{i,j}^m$ roughly gives the "inverse ratio" of the distance between j and i to the average distance between the point j and the points in $C_i^m \cap N_j^m$. If $\eta_{i,j}^m < 3/4$, we think the "average location" of the points in $C_i^m \cap N_j$, denoted by $\bar{k}_{j,i}$, is closer to j than that of i. That is, $\bar{k}_{j,i}$ lies between i and j, and thus, an extrapolation formula for e_j in terms of e_i and $\sum_{k \in C_i^m \cap N_j^m} g_{j,k} e_k^m$ can be applied. When $\eta_{i,j}^m > 2$, we think i is closer to j than that of $\bar{k}_{j,i}$. In this case, we use an interpolation formula instead. Otherwise, we think $\bar{k}_{j,i}$ is very close to j and we should just use the average formula $\sum_{k \in C_i^m \cap N_j^m} g_{j,k} e_k^m$ to approximate e_j.

In summary, we use the following "geometric" interpolation formulae.

(1) For $j \in S_i^m \cap F^m$, we have

$$e_j^m = \begin{cases} 2\sum_{k \in C_i^m} g_{j,k}^m e_k^m - e_i^m, & \text{if } \eta_{i,j}^m < 3/4, \zeta_{i,j} \geq 1/2 \text{ and } a_{i,j}^m < 0 \\ \frac{1}{2}(\sum_{k \in C_i^m} g_{j,k}^m e_k^m + e_i^m), & \text{if } \eta_{i,j}^m > 2, \zeta_{i,j}^m \geq 1/2 \text{ and } a_{i,j}^m < 0 \\ \sum_{k \in C_i^m} g_{j,k}^m e_k^m, & \text{otherwise.} \end{cases} \tag{3.6}$$

(2) For $j \in W_i^m$, we have

$$e_j^m = \begin{cases} e_i^m, & \text{if } C_i^m \cap S_j^m = \phi,\ a_{i,j}^m < 0 \\ -e_i^m, & \text{if } C_i^m \cap S_j^m = \phi,\ a_{i,j}^m > 0 \\ 2\sum_{k \in C_i^m} g_{j,k}^m e_k^m - e_i^m, & \text{if } C_i^m \cap S_j^m \neq \phi,\ \zeta_{i,j}^m \geq 1/2 \text{ and } a_{i,j}^m < 0 \\ \sum_{k \in C_i^m} g_{j,k}^m e_k^m, & \text{otherwise.} \end{cases} \tag{3.7}$$

The convergence proof for this improved AMG method was given in [8] when A^m is symmetric positive definite. Many numerical examples support the improvement of this "geometric" interpolation formula [8]. In Table 1, we compare the convergent rate ρ_A of two AMG methods: Ruge and Stüben's formulae (3.4) (3.5) and our geometric interpolation formulae (3.6) (3.7). This result also supports such improvement. There are other interpolation formulae, for reader's reference, please see [11].

4. Krylov Subspace Acceleration. The Krylov subspace method [5, 15] is an acceleration technique for general iteration methods. Basically, it uses extrapolation to accelerate convergence rate. It is particularly suitable to accelerate our fixed point iteration, because the functional (1.1) is convex, or equivalently, the operator of the corresponding Euler-Lagrange equation (1.2) is monotone. We illustrate this acceleration procedure below. First, we choose two parameters K and s, with $K \leq s$. The Krylov subspace acceleration is perform after every s steps of fixed point iterations as the follows. For integer $n > 0$, let

$$U^{new} = U^{ns} + \sum_{k=1}^{K} \alpha_k (U^{ns+1-k} - U^{ns-k}). \tag{4.1}$$

where the coefficients α_k are chosen such that the residual R^{new} for U^{new} is minimum in L_2 norm, i.e.,

$$\min_{\alpha_1, \cdots, \alpha_K} (R^{new}, R^{new}). \tag{4.2}$$

We then reset U^{ns} to be U^{new}.

Noticing

$$R^{new} = R^{ns} + \sum_{k=1}^{K} \alpha_k (R^{ns+1-k} - R^{ns-k}), \tag{4.3}$$

the coefficients $(\alpha_1, \cdots, \alpha_K)$ can be found easily. For instance, for $K = 1$, α_1 is

$$\alpha_1 = \frac{-(R^{ns},\ R^{ns} - R^{ns-1})}{(R^{ns} - R^{ns-1},\ R^{ns} - R^{ns-1})} \tag{4.4}$$

Remark. The acceleration formula with $K = 1$ is the same as the one given by Brandt and Mikulinsky in [5] when the iteration is linear. Indeed,

$$\alpha_1 = \frac{-\lambda}{\lambda - 1} \text{ with } \lambda = \frac{(R^{ns},\ R^{ns} - R^{ns-1})}{(R^{ns-1},\ R^{ns} - R^{ns-1})}.$$

And λ here is approximately the largest eigenvalue of the iteration matrix M. The reason is shown below.

Suppose the eigenvalues of M are $\lambda_1, \lambda_2, \cdots, \lambda_m = \lambda$ with the corresponding eigenvectors $v_1, v_2, \cdots, v_m$. Then

$$R^0 = r_1^0 v_1 + r_2^0 v_2 + ... + r_m^0 v_m,$$

$$R^n s = \lambda_1^{ns} r_1^0 v_1 + \lambda_2^{ns} r_2^0 v_2 + ... + \lambda_m^{ns} r_m^0 v_m \approx \lambda_m^{ns} r_m^0 v_m \equiv \lambda^{ns} r_n^0 v_m.$$

$$\frac{(R^{ns},\ R^{ns} - R^{ns-1})}{(R^{ns-1},\ R^{ns} - R^{ns-1})} \approx \frac{(\lambda^{ns} r_m^0 v_m, \quad \lambda^{ns-1}(\lambda - 1) r_m^0 v_m)}{(\lambda^{ns-1} r_m^0 v_m, \quad \lambda^{ns-1}(\lambda - 1) r_m^0 v_m)} = \lambda.$$

5. Initialization for the Nonlinear Iteration — a grid continuation procedure.

A good initial guess u^0 (2.6) is important to reduce the number of iterations in the above fixed point method. We shall use a grid continuation procedure, an idea from geometric multigrid method, to construct our initial guess. First, we start at the coarsest grid 4 × 4 grid by performing the fixed point iteration (2.6) few times until the residual is reduced to 10^{-2} of the original one. Here, the noisy data z is restricted to this coarsest grid through a usual restriction operator (5.1):

$$u_{l,k}^c = \frac{1}{4}(u_{2l-1,2k-1}^f + u_{2l-1,2k}^f + u_{2l,2k-1}^f + u_{2l,2k}^f). \tag{5.1}$$

Next, we interpolate the computed result to the next level grid (8×8) by the following interpolation formula: At (x_{2l-1}, y_{2k-1}), we take the same values from the coarse grid. At (x_{2l}, y_{2k}), we use

$$u_{2l,2k}^f = \frac{1}{16}(9u_{l,k}^c + 3u_{l+1,k}^c + 3u_{l,k+1}^c + u_{l+1,k+1}^c). \tag{5.2}$$

For the rest grid points, we use bilinear interpolation. This interpolated U is used as the initial guess for the fixed point iteration at this grid level 8 × 8. We continue this process until we reach the finest grid 256 × 256. At every level, we use our AMG method and the Krylov acceleration to solve the systsm (2.6).

In this grid continuation procedure, a natural diffusion regularization is introduced implicitly, because there is no diffusion degeneracy problem at coarse grid at all. Thus, we can choose a very small β from the beginning.

6. Numerical Experiments and Discussions.

In the numerical experiments below, we take the blur operator K to be the identity matrix (i.e. $K = I$). Two benchmark model problems are considered here [7, 3]. The original images (denoted by u^e, see Figure 1) have 256×256 pixels. Each pixel has value in $[0, 255]$. A Gaussian distribution with mean 0 and variance σ (to be determined later) is added to the original images and the resulting noisy images (denoted by z) are displayed in Figure 2, respectively. By definition, we have $\|u^e - z\|_{L_2} = \sigma$. We choose σ such that the noise-to-signal ratios of these two noisy images are

$$\frac{\|u^e - z\|_{L_2}}{\|u^e\|_{L_2}} = \begin{cases} 0.21 & \text{for model I} \\ 0.28 & \text{for model II.} \end{cases} \tag{6.1}$$

The values of grey level of the original image is $120, 10$ for model I, and $230, 215, 140, 0$ for model II. The value of σ is approximate 35. At the end of this section, we shall also demonstrate denoised results for large noise-to-signal ratios.

During the denoising process, following T. Chan et al's experiment [7], we take $\alpha = 1.18$, and $\beta = 0.01,\ 0.001$. Due to a good initialization, we can further take $\beta = 10^{-32}$ in our experiment.

In the AMG procedure, we apply the simple V-cycle and use the Gauss-Seidel iteration as the smoother.

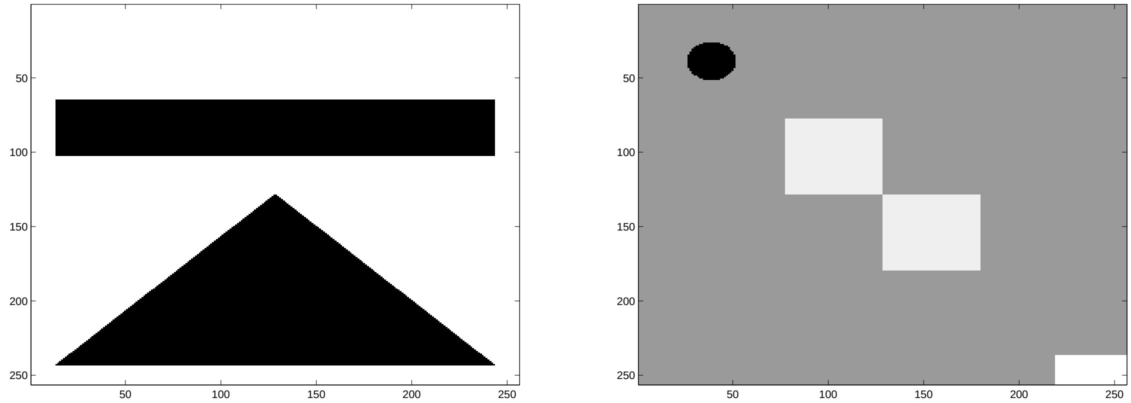

FIG. 6.1. *Original images of model I (left) and model II (right)*

6.1. Normalized residual.

An important issue in image restorations is to choose a quantity to measure the quality of improvement. It is used as a stopping criterion for the fixed point iteration. Usually, the residual of the system (2.6) is chosen. But, a normalization is needed. Namely, we should use $D^{-1}(Re)$ as the normalized residual. Here, (Re) is the residual of the system (2.6) and D is the corresponding diagonal matrix. The reason for this normalization is the follows. Due to the fact that the diffusion coefficient is very large in smooth region, we observe that the unnormalized residual is very large in those components where u is smooth (thus, no more denoising is needed), and is relatively small in those where u is less smooth (thus, either it has a jump or it needs further denoising). A normalization will cure this imbalance. Numerical experiments below demonstrates that this quantity is able to measure the improvement of the denoising process. From now on, we shall denote this normalized residual by Re.

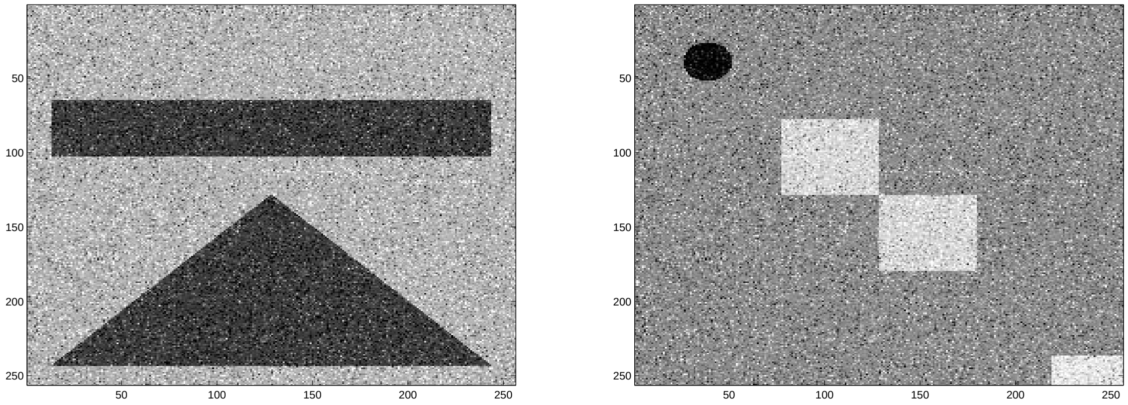

Fig. 6.2. *Noisy images of model I (left) and model II (right). Noise-to-signal ratios: 0.21 (model I), 0.28 (model II).*

6.2. Fixed point iteration with AMG method. In the first set of numerical experiments, we apply the fixed point method to the model images at the finest grid directly. No particular preparation of initial data and nor the Krylov acceleration are adopted at this moment. In each fixed point iteration, only one V-cycle of the AMG method is applied for solving the corresponding linear system. There is no need to have more iteration because the dominant error is from the outer iteration. The stopping criterion for the fixed point iteration in this paper is a relative decrease of the residual by a factor of 10^{-4} for model I and of 10^{-5} for model II, namely,

$$\frac{\|Re^N\|_{L_2}}{\|Re^0\|_{L_2}} \leq \begin{cases} 10^{-4}, & \text{for Model I,} \\ 10^{-5}, & \text{for Model II.} \end{cases}$$

In AMG, we adopt two interpolation formulae: AMG I uses Ruge and Stüben's formula (3.4) (3.5), whereas AMG II uses (3.6) and (3.7). Table 1 is the comparison of the convergent factor ρ_A of these two AMG algorithms, i.e. the ratio of the residual after and before the AMG. The result shows that our AMG algorithm is superior to the Ruge and Stüben's AMG in this denoising test.

iteration step	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
AMG I	0.306	0.006	0.088	0.079	0.097	0.076	0.072	0.068	0.064	0.061	0.058	0.055	0.052	0.052	0.051	0.051	0.050	0.050	0.052	0.055	0.056	0.058
AMG II	0.209	0.009	0.088	0.069	0.064	0.063	0.058	0.052	0.049	0.048	0.045	0.044	0.043	0.043	0.043	0.047	0.051	0.049	0.052	0.053	0.054	0.053

Table 6.1

ρ_A, the convergence factor of the two AMG methods in each fixed point iteration. AMG I is the Ruge-Stüben's formula (3.4) and (3.5), whereas AMG II uses (3.6) and (3.7). This result is for the Model I with $\beta = 0.01$.

Next, we compare our method (fixed-point method + AMG I) for Model I with that of the primal-dual Newton method [7]. Table 2 is the number of fixed point iterations (denoted by N) and the "CPU" time (on Sparc 10 machine) needed in our method. In Figure 3, the solid line on the left figure is the log of the residual versus the iteration steps. Comparing these results with those in [7], we observe that the number of outer iteration is about the same in both methods, whereas the number of inner iteration is just one in our method versus 4 to 5 in their method. The reason is that the linear equation in fixed point method is much easier to solve than that of Newton's method.

Model	β	N	CPU time (in second)
I	0.01	22	79.60
	0.001	22	84.11
II	0.01	48	165.97
	0.001	49	172.6

TABLE 6.2

Number of fixed point iterations N and the CPU time (on Sprac 10) needed to achieve a reduction of residual by a factor of 10^{-4} *for Model I and of* 10^{-5} *for Model II.*

6.3. Improvement by the Krylov Acceleration. The convergence of the fixed point iteration above can further be improved by the Krylov acceleration method. In the application of Krylov acceleration, we choose the parameter $s = 4$, i.e. we apply the Krylov acceleration every four fixed point iterations. The parameter K is taken to be 1 or 2. The result is given in Figure 3. The total number of iterations is reduced to about 50%. The overhead is low, because only simple algebraic operations are needed. The results demonstrate that the Krylov acceleration method is very efficient to accelerate the convergence of our fixed point method.

K	Model	β	N	CPU time (in second)
1	I	0.01	12	44.14
		0.001	12	44.34
	II	0.01	21	74.48
		0.001	21	75.53
2	I	0.01	12	44.64
		0.001	12	44.82
	II	0.01	21	74.72
		0.001	20	71.26

TABLE 6.3

Number of fixed point iterations N needed with Krylov acceleration. Comparing with Table 2, the improvement is almost double.

6.4. Further improvement by a good preparation of initial data. In previous numerical experiments, the initial data is 0. We can have less nonlinear iterations if we start from a good initial guess. As mentioned in previous section, such an initial guess is obtained by interpolation from an approximate solution at coarse grid successively. We start from 4×4 grid. The stopping criterion at each grid level is the 10% reduction of the residual. Table 5 is the number of iteration and CPU time with this initialization. There, C_i, C_f and C_{total} are the CPU times of the initialization, the fixed point iteration and the total time, respectively. We

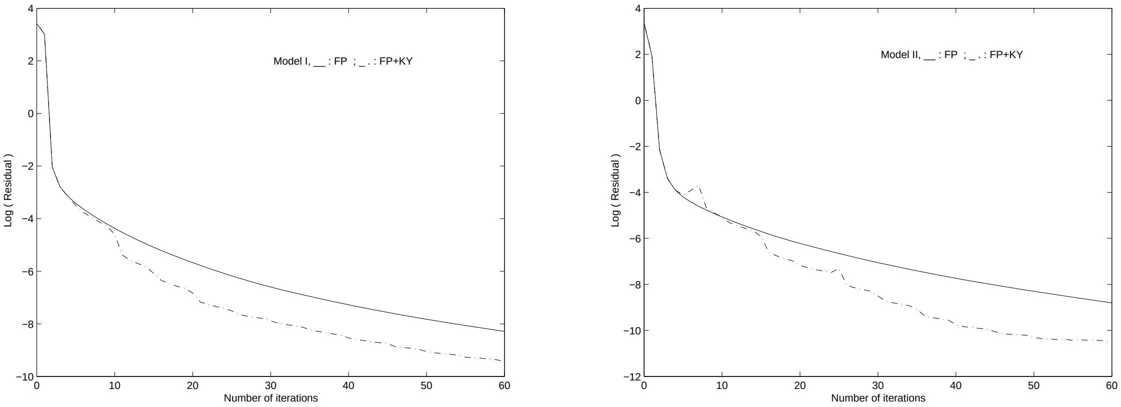

FIG. 6.3. *Log of residuals versus iteration steps for the fixed point method with (dash line) and without (solid line) the Krylov acceleration for Model I (left) and Model II (right)*

observe that the initialization is about 10% of the total work for Model I and 7% for Model II. Comparing Table 3 and Table 5, we observe this initialization gives about 20% improvement. What a surprise is that we are allowed to choose a very small regularization coefficient β, say $\beta = 10^{-32}$, the smallest positive machine number. The overall method still converges. Without a good initialization, the fixed point method converges slowly for $\beta < 10^{-4}$ [10]. Figure 3 is the denoised images with $\beta = 10^{-32}$.

Model	β	N	C_i	C_f	C_{total}
I	0.01	9	3.16	32.46	35.62
	0.001	9	3.18	32.68	35.86
	1.0×10^-32	9	3.20	33.11	36.31
II	0.01	17	4.51	56.87	61.38
	0.001	17	4.48	57.75	62.23
	1.0×10^-32	17	4.59	59.15	63.74

TABLE 6.4

CPU time by using the combination of the fixed point method, AMG algorithm, Krylov acceleration and good preparation of initial guess. The improvement of the good initialization is about 40%. It can also allow us to choose a very small β, the smallest positive machine number.

6.5. Denoised results for images with large noise-to-signal ratios . Finally, we test the denoising quality for images with large noise-to-signal ratios. Figure 5 is the noised images. The noise-to-signal ratios are respectively 2.03 for model I and 2.01 for model II. The corresponding values of noise are between −200 to 200 for model I, and between −300 to 300 for model II. Figure 6 is the denoised images. The denoised image of model II looks darker due to the rescaling of the matlab graphic tool. The denoised results are satisfactory.

In conclusion, numerical experiments demonstrate that our algorithm, which combines the fixed point method, an improved AMG method, the Krylov acceleration and a good initialization, is efficient and robust.

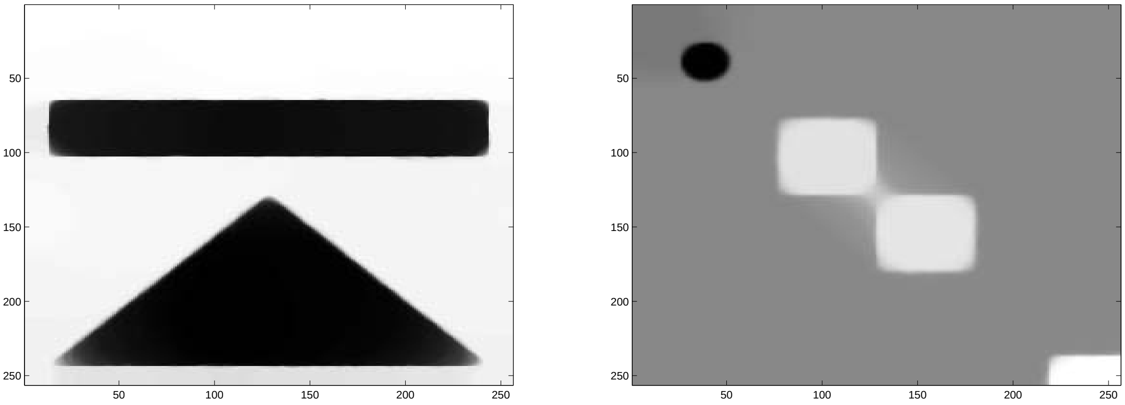

FIG. 6.4. *Restored images of Model I(left) and Model II (right)*

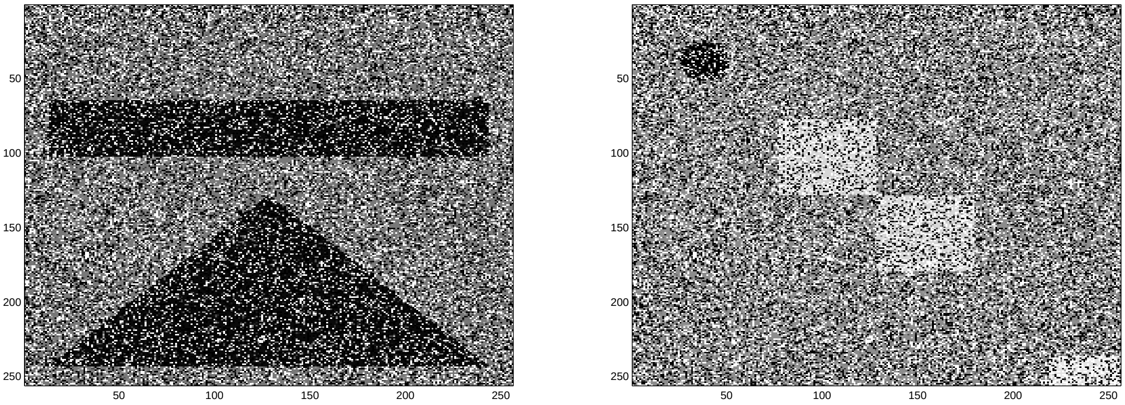

FIG. 6.5. *Noisy images of Model I and II. Noise-to-signal ratios: 2.03 (model I) and 2.01 (model II).*

Acknowledgement. We thank referees for telling us useful references.

REFERENCES

[1] R. Acar and C. R. Vogel, Analysis of total variation penalty methods for ill-psoed problems, Inverse Problems, 10(1994), pp 1217-1229.

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

[3] C. A. Z. Barcelos, and Y. Chen, Heat flow and related minimization problem in image restoration, Computers and Mathematics with Applications, 39(2000), pp 81-97.

[4] G. Barles, and P. E. Souganidis, Convergence of approximation schemes for fully nonlinear second order equations, Asymptotic Analysis 4(1991), pp 271-283.

[5] A. Brandt, and V. Mikulinsky, On recombining iterants in multigrid algorithms and problems with small islands, SIAM Sci. Comput., 16(1995), pp 20-28.

[6] R. Chan, T. Chan, and H. Zhou, Advanced signal processing algorithms, In Proceedings of the International Society of Photo-Optical Instrumentation Engineers, F. Luk, ed., SPIE, 1995, pp 314-325.

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

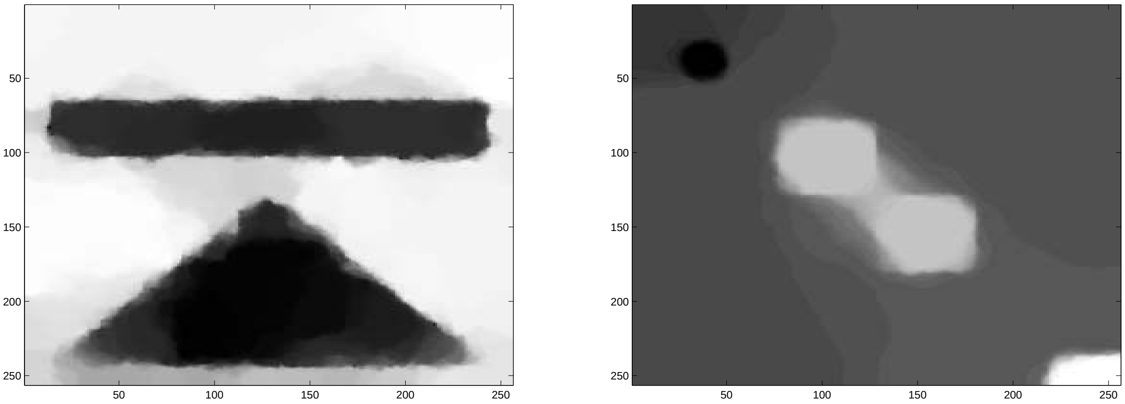

FIG. 6.6. *Denoised images with large noise-to-signal ratios: 2.03 (model I) and 2.01 (model II).*

[8] Q. Chang, Yaushu Wong and H. Fu, J. Comput. Phys., 125(1996), pp.279-292.

[9] Q. Chang, S. Ma and G. Lei, J. of Computer Math., 70(1999), pp. 539-552.

[10] D. Dobson and C. R. Vogel, Convergence of an iterative method for total variation denoising, SIAM J. Numerical Analysis, 34 (1997) p. 1779-1791.

[11] V.E. Henson and P.S. Vassilevski, Element-free AMGe: General algorithms for computing interpolation weights, SIAM J. Sci. Comput. 23 (2001) p. 629-650.

[12] Y. Li and F. Santosa, An affine scaling algorithm for minimizing total variation in image enhancement, Tech. Report 12/94, Center for Theory and Simulation in Science and Engineering, Cornell University, 1994.

[13] S. McCormick, Multigrid Methods, SIAM, Philadelphia, 1987.

[14] M. E. Oman, Fast multigrid techniques in total variation-based image reconstruction, to appear in Preliminary Proceedings of the 1995 Copper Mountain Conference on Multigrid Methods.

[15] C. W. Oosterlee, and T. Washio, Krylov subspace acceleration of nonlinear multigrid with application to recirculating flows, SIAM J. Sci. Comput., 21(2000), pp 1670-1690.

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

[17] J. Ruge and K. St*ü*ben, Algebraic Multigrid, in Multigrid Methods, (S. F. McCormick, ed.) 4, SIAM, Philadephia, (1987) pp. 73-130.

[18] P.S. Vassilevski and J. G.Wade, A comparison of multilevel methods for total variation regularization, Electric Transaction on Numerical Analysis, 6 (1997), pp 255- 280.

[19] C. R. Vogel, A multigrid method for total variation-based image denoising, in Computation and Control IV, Progr. Systems Control Theory 20, Birkhauser, Boston, MA, 1995, pp 323-331.

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

[21] Y. Zhou, Applications of discrete functional analysis to the finite difference method, International Academic Publishers, Beijing, 1991.