Electronic Transactions on Numerical Analysis.
Volume 26, pp. 299-311, 2007.

ISSN 1068-9613.

ETNA
Kent State University
etna@mcs.kent.edu

ITERATIVE METHODS FOR SOLVING THE DUAL FORMULATION ARISING FROM IMAGE RESTORATION*

TONY F. CHAN†, KE CHEN‡, AND JAMYLLE L. CARTER§

Abstract. Many variational models for image denoising restoration are formulated in primal variables that are directly linked to the solution to be restored. If the total variation related semi-norm is used in the models, one consequence is that extra regularization is needed to remedy the highly non-smooth and oscillatory coefficients for effective numerical solution. The dual formulation was often used to study theoretical properties of a primal formulation. However as a model, this formulation also offers some advantages over the primal formulation in dealing with the above mentioned oscillation and non-smoothness. This paper presents some preliminary work on speeding up the Chambolle method [J. Math. Imaging Vision, 20 (2004), pp. 89–97] for solving the dual formulation. Following a convergence rate analysis of this method, we first show why the nonlinear multigrid method encounters some difficulties in achieving convergence. Then we propose a modified smoother for the multigrid method to enable it to achieve convergence in solving a regularized Chambolle formulation. Finally, we propose a linearized primal-dual iterative method as an alternative stand-alone approach to solve the dual formulation without regularization. Numerical results are presented to show that the proposed methods are much faster than the Chambolle method.

Key words. image restoration, nonlinear partial differential equations, singularity, nonlinear iterations, Fourier analysis, multigrid method

AMS subject classifications. 68U10, 65F10, 65K10

1. Introduction. Image restoration is the most basic problem in the long list of image processing problems. Often such restored images are used for further tasks such as image segmentation and matching [1, 11].

The best-known model for denoising is the following primal formulation by Rudin-Osher-Fatemi ROF [18]

$$\min_u \int_\Omega \alpha|\nabla u| + \frac{1}{2}(Ku - z)^2 dxdy, \tag{1.1}$$

or its Euler-Lagrange form for the denoising case $K = I$ of interest to us

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

where $z \in \mathbb{R}^2(\Omega)$ is the observed image and u is our desired image (with homogeneous Neumann boundary conditions) to be restored. Here and in (1.1), the term with $|\nabla u| = \sqrt{u_x^2 + u_y^2}$ defines the total variation (TV) of function u. In practice, only a discrete quantity $z \in \mathbb{R}^{n\times n}$ will be available. In (1.2), one assumes that $|\nabla u| \neq 0$; to allow generality, a replacement equation with $|\nabla u|_\beta = \sqrt{|\nabla u|^2 + \beta}$ for some $\beta > 0$ may be considered,

$$-\alpha\nabla\cdot\left(\frac{\nabla u}{|\nabla u|_\beta}\right) + u - z = 0. \tag{1.3}$$

*Received August 3, 2006. Accepted for publication May 11, 2007. Recommended by R. Plemmons.

†Department of Mathematics, University of California, Los Angeles, CA 90095-1555, USA (chan@math.ucla.edu). This work is supported in part by the Office of Naval Research ONR N00014-03-1-0888 and ONR N00014-06-1-0345, and by the National Institutes of Health NIH U54-RR021813.

‡Department of Mathematical Sciences, University of Liverpool, Peach Street, Liverpool L69 7ZL, UK (k.chen@liverpool.ac.uk). This work is supported in part by the UK Leverhulme Trust RF/9/RFG/2005/0482. (For correspondence).

§Department of Mathematics, San Francisco State University, 1600 Holloway Ave, San Francisco, CA 94132, USA (jlc@sfsu.edu).

ETNA
Kent State University
etna@mcs.kent.edu

Other non-minimization models may be proposed directly in the form of a differential equation, e.g., the Perona-Malik [17] model for the same denoising problem takes the form

$$\frac{\partial u}{\partial t} = \nabla \cdot [D(|\nabla u|)\nabla u]\,,$$

where $u = u(x, y, t)$, $u(x, y, 0) = z$, and the nonlinear diffusive coefficient D is designed to restore sharp edges in u. It turns out that these two kinds of models are closely related [13]; see also [9, 19] for more sophisticated models than ROF [18] for denoising. Here we are mainly concerned with the ROF model.

There exist many numerical methods for solving both (1.1) and (1.2); see [6, 7, 8] for detailed surveys and the references therein. Here we shall address the question of how to solve the dual equation [5] (details are discussed in the next section):

$$-\nabla(\alpha \mathrm{div}(\boldsymbol{\omega}) - z) + |\nabla(\alpha \mathrm{div}(\boldsymbol{\omega}) - z)|\,\boldsymbol{\omega}_{i,j} = 0, \tag{1.4}$$

where $\boldsymbol{\omega} = (\omega_1, \omega_2)$ is the dual variable and the restored solution will be given by $u = z - \alpha\mathrm{div}(\boldsymbol{\omega})$. The clear advantage of the dual formulation (1.4) over the primal equation (1.3) is that no such extra regularizing parameter β is needed and the dual equation appears more amenable—however, despite the appearance, as we shall see, the latter equation is deceptively challenging as far as finding efficient solvers is concerned. There exist few studies of this dual equation. In [16], a non-smooth Newton method is considered but the achieved convergence is not yet quadratic. A related dual problem (for a different but slightly easier functional than the ROF model) has been studied recently in [14].

2. Review of the dual and related formulations. Casting the primal formulation (1.2) into a dual formulation is a main approach used to study the theoretical properties of u. Here our aim is to address methods of solving the dual formulations effectively. We first review some previous studies and then present their connections. Although equation (1.4) is commonly known as the Chambolle's dual formulation, three other papers [4, 10, 15] are closely connected to this formulation. We now review their connections for readers' convenience.

From Meyer's G-space theory [15], the solution to model (1.2) lies in the G-space, i.e.,

$$u - z \in \mathcal{G} = \{f \mid \|f\|_* < \infty,\ f(x, y) = \nabla \cdot \mathbf{g} = \partial_x g_1(x, y) + \partial_y g_2(x, y),\ g_1, g_2 \in L^\infty\},$$

where $\mathbf{g} = (g_1, g_2)$ and $\|f\|_*$ is defined as the lower bound of all $L^\infty(\Omega)$ norms of the function $|g(x, y)| = \sqrt{g_1^2 + g_2^2}$. Therefore it is valid to assume that

$$u = z + \gamma\nabla\boldsymbol{\omega} \tag{2.1}$$

for some γ; it turns out that such a parameter is precisely $-\alpha$ to coincide with [4, 5].

Upon introducing the dual variable $\boldsymbol{\omega}$ to replace the TV-term by applying the equality

$$|\nabla u| = \max_{|\boldsymbol{\omega}|\le 1} \boldsymbol{\omega} \cdot \nabla u$$

to the ROF model (1.1) and interchanging the min and max, Carter [4] derived (as in (2.1))

$$u = z - \alpha\nabla\boldsymbol{\omega},$$

and then she proposed to numerically solve

$$\max_{|\boldsymbol{\omega}|\le 1} \int_\Omega (\nabla \cdot \boldsymbol{\omega})z - \frac{\alpha}{2}(\nabla \cdot \boldsymbol{\omega})^2 dxdy. \tag{2.2}$$

ETNA
Kent State University
etna@mcs.kent.edu

Various unilevel (optimization) relaxation methods are considered for solving (2.2) in [4]. It was not clear which method comes out as more robust than others for (2.2).

In order to solve (1.1), Chambolle [5] starts by looking for the projection $\pi_\alpha(z) = \alpha \mathrm{div}(\boldsymbol{\omega})$, with $u = z - \pi_\alpha(z)$ in mind, from

$$\min_{\boldsymbol{\omega}} \|\alpha \mathrm{div}(\boldsymbol{\omega}) - z\|_2^2, \qquad |\boldsymbol{\omega}_{i,j}| \le 1 \; \forall (i,j), \tag{2.3}$$

where $|\boldsymbol{\omega}_{i,j}| = \sqrt{(w_1)_{i,j}^2 + (w_2)_{i,j}^2}$. It remains to solve the unconstrained KKT problem

$$-\nabla \left(\alpha \mathrm{div}(\boldsymbol{\omega}) - z\right)_{i,j} + \delta_{i,j} \boldsymbol{\omega}_{i,j} = 0.$$

Recognizing either $\delta_{i,j} > 0$, $|\boldsymbol{\omega}_{i,j}| = \sqrt{\omega_1^2 + \omega_2^2} = 1$ or $\delta_{i,j} = 0$, $|\boldsymbol{\omega}_{i,j}| < 1$, Chambolle [5] finds $\delta_{i,j} = \left|\nabla \left(\mathrm{div}(\boldsymbol{\omega}) - \frac{z}{\alpha}\right)\right|_{i,j}$ and presents the dual formulation in the form

$$-\left(\nabla(\mathrm{div}(\boldsymbol{\omega}) - \frac{z}{\alpha})\right)_{i,j} + \left|\nabla(\mathrm{div}(\boldsymbol{\omega}) - \frac{z}{\alpha})\right|_{i,j} \boldsymbol{\omega}_{i,j} = 0, \tag{2.4}$$

or at each pixel (i,j)

$$\begin{cases} \dfrac{\partial^2 \omega_1}{\partial x^2} + \dfrac{\partial^2 \omega_2}{\partial x \partial y} - \dfrac{1}{\alpha}\dfrac{\partial z}{\partial x} = \sqrt{\left(\dfrac{\partial^2 \omega_1}{\partial x^2} + \dfrac{\partial^2 \omega_2}{\partial x \partial y} - \dfrac{1}{\alpha}\dfrac{\partial z}{\partial x}\right)^2 + \left(\dfrac{\partial^2 \omega_2}{\partial y^2} + \dfrac{\partial^2 \omega_1}{\partial y \partial x} - \dfrac{1}{\alpha}\dfrac{\partial z}{\partial y}\right)^2}\,\omega_1, \\ \dfrac{\partial^2 \omega_2}{\partial y^2} + \dfrac{\partial^2 \omega_1}{\partial y \partial x} - \dfrac{1}{\alpha}\dfrac{\partial z}{\partial y} = \sqrt{\left(\dfrac{\partial^2 \omega_1}{\partial x^2} + \dfrac{\partial^2 \omega_2}{\partial x \partial y} - \dfrac{1}{\alpha}\dfrac{\partial z}{\partial x}\right)^2 + \left(\dfrac{\partial^2 \omega_2}{\partial y^2} + \dfrac{\partial^2 \omega_1}{\partial y \partial x} - \dfrac{1}{\alpha}\dfrac{\partial z}{\partial y}\right)^2}\,\omega_2. \end{cases}$$

Using a semi-implicit time-marching scheme, the so-called Chambolle method [5] proposes to solve the semi-implicit scheme

$$\frac{\boldsymbol{\omega}_{i,j}^{(k+1)} - \boldsymbol{\omega}_{i,j}^{(k)}}{\tau} = \left(\nabla(\mathrm{div}(\boldsymbol{\omega}^{(k)}) - \frac{z}{\alpha})\right)_{i,j} - \left|\nabla(\mathrm{div}(\boldsymbol{\omega}^{(k)}) - \frac{z}{\alpha})\right|_{i,j} \boldsymbol{\omega}_{i,j}^{(k+1)} \tag{2.5}$$

or

$$\boldsymbol{\omega}_{i,j}^{(k+1)} = \frac{\boldsymbol{\omega}_{i,j}^{(k)} + \tau \left(\nabla(\mathrm{div}(\boldsymbol{\omega}^{(k)}) - \frac{z}{\alpha})\right)_{i,j}}{1 + \tau \left|\nabla(\mathrm{div}(\boldsymbol{\omega}^{(k)}) - \frac{z}{\alpha})\right|_{i,j}}.$$

It is the purpose of this paper to address alternative methods for solving (2.4).

2.1. Equivalence of the Carter formulation to the Chambolle formulation. To relate (2.2) to (2.3), we change the sign of the functional in (2.2) to introduce the equivalent min problem and neglect the last term (not depending on $\boldsymbol{\omega}$),

$$\min_{|\boldsymbol{\omega}| \le 1} \int_\Omega \left[\frac{\alpha^2}{2}(\nabla \cdot \boldsymbol{\omega})^2 - \alpha(\nabla \cdot \boldsymbol{\omega}) z\right] dxdy = \min_{|\boldsymbol{\omega}| \le 1} \frac{1}{2} \int_\Omega \left[(\alpha \nabla \cdot \boldsymbol{\omega} - z)^2 - z^2\right] dxdy,$$

which is clearly equivalent to the following problem:

$$\min_{|\boldsymbol{\omega}|} \|\nabla \cdot \boldsymbol{\omega} - z\|_2^2, \qquad |\boldsymbol{\omega}| \le 1.$$

This is in turn the same as the Chambolle equation (2.3), so, following the same argument, we arrive at

$$-\left(\nabla(\mathrm{div}(\boldsymbol{\omega}) - \frac{z}{\alpha})\right) + \left|\nabla(\mathrm{div}(\boldsymbol{\omega}) - \frac{z}{\alpha})\right| \boldsymbol{\omega} = 0.$$

ETNA
Kent State University
etna@mcs.kent.edu

2.2. Equivalence of the primal-dual method [10] to the Chambolle formulation. To solve (1.3), Chan-Golub-Mulet [10] proposed the primal-dual formulation

$$\begin{cases} -\alpha\nabla\cdot\omega + u - z = 0, \\ \omega\sqrt{|\nabla u|^2+\beta} - \nabla u = 0, \end{cases} \tag{2.6}$$

in two variables (u, ω) after introducing in (1.3) the dual variable by $\omega{=}\nabla u/\sqrt{|\nabla u|^2{+}\beta}$ for some small $\beta > 0$. In fact, when $|\nabla u| = \beta = 0$, equation (2.6) is still valid and is equivalent to (1.2) when $|\nabla u| \neq 0$ so we may let $\boldsymbol{\omega} = -\omega$ and rewrite (2.6), for the purpose of modeling u, as

$$\begin{cases} \alpha\mathrm{div}(\boldsymbol{\omega}) + u - z = 0, \\ \boldsymbol{\omega}|\nabla u| + \nabla u = 0, \end{cases} \tag{2.7}$$

where one observes $\nabla\cdot\boldsymbol{\omega} = \mathrm{div}(\boldsymbol{\omega})$. One can eliminate u in the second equation using the first equation, i.e. $u = z - \alpha\mathrm{div}(\boldsymbol{\omega})$, to derive that

$$\boldsymbol{\omega}|\nabla(z-\alpha\mathrm{div}(\boldsymbol{\omega}))| + \nabla(z-\alpha\mathrm{div}(\boldsymbol{\omega})) = 0, \text{ or } -\nabla(\alpha\mathrm{div}(\boldsymbol{\omega})-z) + |\nabla(z-\alpha\mathrm{div}(\boldsymbol{\omega}))|\boldsymbol{\omega} = 0,$$

which is evidently the same as the Chambolle formulation (2.4) after scaling by α.

3. Nonlinear multigrid for the Chambolle's dual formulation. We now turn to the main aim of this paper on developing a multigrid algorithm for the Chambolle formulation (2.4):

$$-\nabla\left(\alpha\mathrm{div}(\boldsymbol{\omega}) - z\right)_{i,j} + \left|\nabla(\mathrm{div}(\boldsymbol{\omega}) - \frac{z}{\alpha})\right|_{i,j}\boldsymbol{\omega}_{i,j} = 0.$$

It turns out that this equation cannot be easily solved by a multigrid method as shown below. Simple smoothers such as Jacobi and Gauss-Seidel do not work; the experts [3] are unfortunately not surprised about this as (2.4) is degenerate. Then in the next section, after some analysis, we shall present alternative algorithms.

To proceed, it is convenient to denote the discretized version of this equation in the operator notation by

$$\mathcal{N}_1\mathbf{w}_1 = 0, \tag{3.1}$$

where $\mathbf{w}_1 = [(\omega_1)_{1,1}, (\omega_1)_{1,2}, \ldots, (\omega_1)_{n,n}, (\omega_2)_{1,1}, (\omega_2)_{1,2}, \ldots, (\omega_2)_{n,n}]^T$ denotes our solution on the finest level 1 with $n\times n$ pixels. Let $n = 2^{L+2}-1$. We remark that ω_1, ω_2 will have 0 Dirichlet boundary conditions so we shall assume the given image $z \in \mathbb{R}^{n\times n}$ is embedded in a larger domain having a zero boundary condition at all sides. Let a standard coarsening strategy be used to generate a sequence of coarse levels $k = 2, 3, \ldots, L$ with the kth level containing $n_k \times n_k$ pixels with $n_k = 2^{L+2-k} - 1$ so that the coarsest level has $n_L = 2^2 - 1 = 3$. Similarly denote a level k discretization of (2.4) by

$$\mathcal{N}_k\mathbf{w}_k = 0.$$

We use the standard nonlinear multilevel algorithm by Brandt [2], as detailed in Chen [12, Alg. 6.2.8].

Start from an initial guess $\mathbf{w}_1$. Assume we have set up these multigrid parameters:

ν_1 — pre-smoothing steps on each level k (before restriction)
ν_2 — post-smoothing steps on each level k (after interpolation)
γ — multigrid cycles on each level k or the cycling pattern (here $\gamma = 1$).

ETNA
Kent State University
etna@mcs.kent.edu

ALGORITHM 3.1 (Nonlinear Multigrid).

1. *Based on the current approximation* $\mathbf{w}_1$ *to equation (3.1), to do* ℓ *steps of a* γ*-cycling nonlinear multigrid, use for* ℓ *times*

$$NMG(\mathbf{w}_1, g_1, 1).$$

2. *One step of a* γ*-cycling nonlinear multigrid NMG*$(\mathbf{w}_k, g_k, k)$ *proceeds as follows*

$NMG(\mathbf{w}_k, g_k, k)$:
if $k = L$, *then*
— *Solve on level* L *for* $\mathbf{w}_L$ *accurately* , $\mathcal{N}_L \mathbf{w}_L = g_L$
else, on level k*, do*
— *Pre-smoothing* : $\mathbf{w}_k = \mathrm{Relax}_k^{\nu_1}(\mathbf{w}_k, g_k)$,
— *Restrict to the coarse grid,* $\mathbf{w}_{k+1} = R_k^{k+1}\mathbf{w}_k$
— *Compute* $g_{k+1} = R_k^{k+1}(g_k - \mathcal{N}_k \mathbf{w}_k) + \mathcal{N}_{k+1}\mathbf{w}_{k+1}$,
— *Set the initial solution on level* $k+1$ *as* $\widetilde{\mathbf{w}}_{k+1} = \mathbf{w}_{k+1}$,
— *Implement* γ *steps of* $NMG(\widetilde{\mathbf{w}}_{k+1}, g_{k+1}, k+1)$,
— *Add the residual correction,* $\mathbf{w}_k = \mathbf{w}_k + P_{k+1}^k(\widetilde{\mathbf{w}}_{k+1} - \mathbf{w}_{k+1})$,
— *Post-smoothing* : $\mathbf{w}_k = \mathrm{Relax}_k^{\nu_2}(\mathbf{w}_k, g_k)$,
end if k
end one step of $NMG(\mathbf{w}_k, g_k, k)$.

Here we make the usual choice of the restriction operator R_k^{k+1} as the simple injection and operator P_{k+1}^k as the bilinear interpolation. It remains to discuss how to implement $\mathrm{Relax}_k^{\nu}(\mathbf{w}_k, g_k)$ which denotes a relaxation method applying to $\mathcal{N}_k\mathbf{w}_k = g_k$ on level k for ν steps.

Some experiments conducted on relaxation methods show that the following methods are not suitable smoothers: point-wise Jacobi, Gauss-Seidel and the 2×2 block Jacobi method. The natural choice appears to use the Chambolle iterations (2.5).

Take $\nu_1 = \nu_2 = 4$ and use 200 iterations on the coarsest level L. We found that the nonlinear Algorithm 3.1 fails to converge quickly. To exclude the possibility of $\left|\nabla(\mathrm{div}(\boldsymbol{\omega}^{(k)}) - \frac{z}{\alpha})\right|$ being small or zero, we considered modifying the Chambolle iteration (2.5) to

$$\frac{\boldsymbol{\omega}_{i,j}^{(k+1)} - \boldsymbol{\omega}_{i,j}^{(k)}}{\tau} = \left(\nabla(\mathrm{div}(\boldsymbol{\omega}^{(k)}) - \frac{z}{\alpha})\right)_{i,j} - \sqrt{\left|\nabla(\mathrm{div}(\boldsymbol{\omega}^{(k)}) - \frac{z}{\alpha})\right|_{i,j}^2 + \beta}\,\boldsymbol{\omega}_{i,j}^{(k+1)}, \tag{3.2}$$

where $\beta = 10^{-3}$ is used. Again only minor improvements were observed. We show 2 test results in Figs. 3.1 and 3.2, where the top plots use $\beta = 0$ and the bottom $\beta = 10^{-3}$. Clearly there should be reasons why Algorithm 3.1 converges slowly for $\beta = 0$.

4. Convergence rate analysis and modified algorithms. We first use Fourier analysis to gain some insight into the behavior of the Chambolle iterations and the associated Algorithm 3.1. Then this analysis is used to assist us to propose alternative algorithms that would converge.

4.1. Fourier analysis. Recall that the local Fourier analysis (LFA) aims to measure the largest amplification factor in a relaxation scheme [2, 12, 20]. Let the general Fourier component be

$$\Phi_{\alpha,\beta}(x_\ell, y_m) = \exp\left(i\theta_\alpha \frac{x_\ell}{h} + i\theta_\beta \frac{y_m}{h}\right) = \exp\left(\frac{2i\alpha\ell\pi}{n} + \frac{2i\beta m\pi}{n}\right),$$

ETNA
Kent State University
etna@mcs.kent.edu

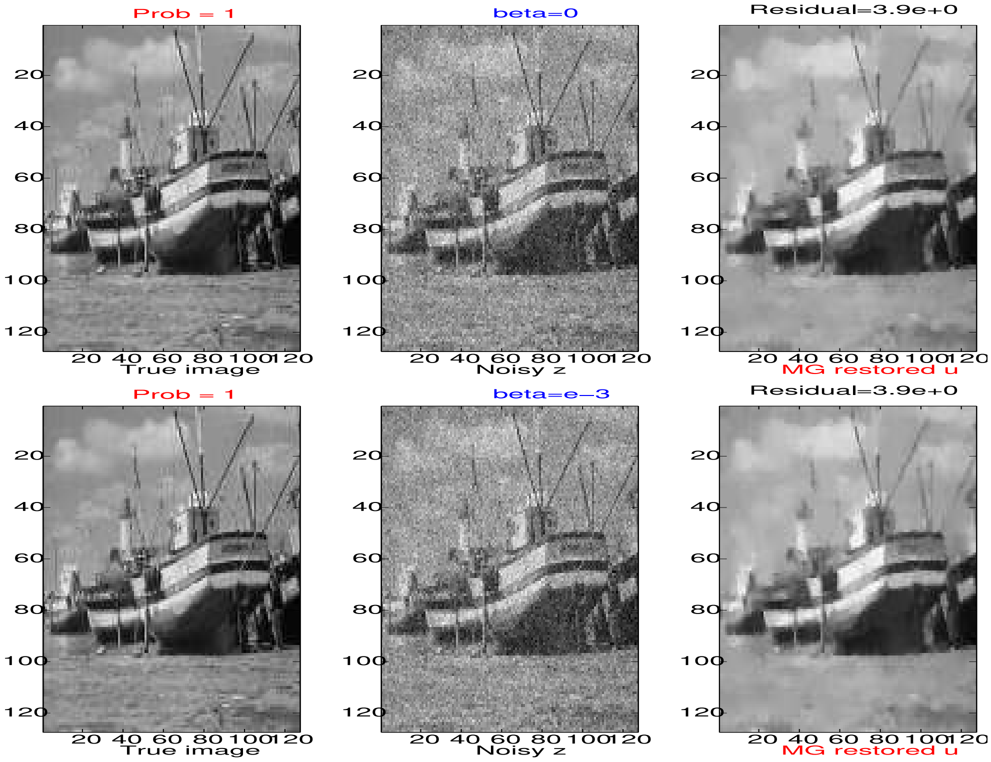

FIG. 3.1. *Test example* 1 *to show the slow convergence of the standard Algorithm 3.1, with the left, the center and the right plots displaying respectively the true image, the noisy image and the restored image by* 40 *cycles of MG. Here* $\alpha = 20$ *and the residuals* $r_{\beta=0} = r_{\beta=10^{-3}} = 3.9$.

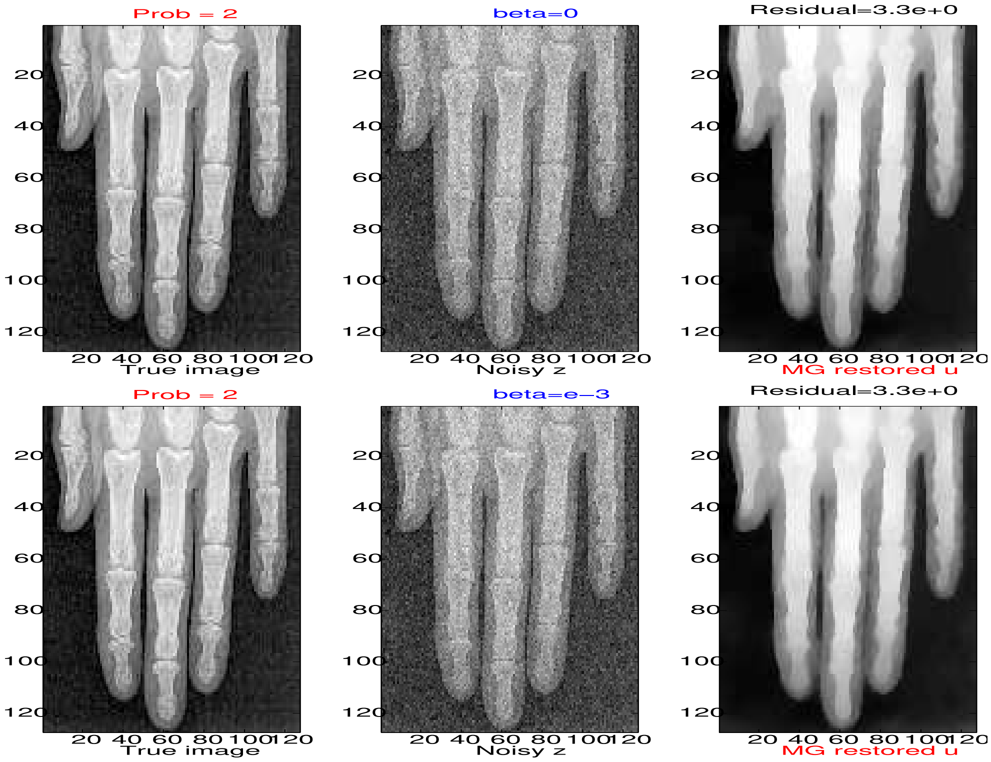

FIG. 3.2. *Test example* 2 *to show the slow convergence of the standard Algorithm 3.1, with the left, the center and the right plots displaying respectively the true image, the noisy image and the restored image by* 40 *cycles of MG. Here* $\alpha = 40$ *and the residuals* $r_{\beta=0} = r_{\beta=10^{-3}} = 3.3$

and define the local error functions by $e_1^{(k)} = \omega_1 - \omega_1^{(k)}$ and $e_2^{(k)} = \omega_2 - \omega_2^{(k)}$. Here $\theta_\alpha = 2\alpha\pi/n, \theta_\beta = 2\beta\pi/n \in [-\pi, \pi]$. Then LFA involves expanding

$$e_1^{(k)} = \sum_{\alpha,\beta=-n/2}^{n/2} (\psi_1^{(k)})_{\alpha,\beta}\Phi_{\alpha,\beta}(x_\ell, y_m), \quad e_2^{(k)} = \sum_{\alpha,\beta=-n/2}^{n/2} (\psi_2^{(k)})_{\alpha,\beta}\Phi_{\alpha,\beta}(x_\ell, y_m)$$

in Fourier components. We shall first estimate the maximum spectral radius $\max_{\alpha,\beta}\rho(\mathcal{A}_{\alpha,\beta})$ where $\mathcal{A}_{\alpha,\beta}$, of size 2×2, denotes the amplification matrix for our smoothing method for the dual system:

$$\begin{bmatrix} (\psi_1^{(k+1)})_{\alpha,\beta} \\ (\psi_2^{(k+1)})_{\alpha,\beta} \end{bmatrix} = \mathcal{A}_{\alpha,\beta} \begin{bmatrix} (\psi_1^{(k)})_{\alpha,\beta} \\ (\psi_2^{(k)})_{\alpha,\beta} \end{bmatrix}.$$

Consider the Chambolle smoothing iterations with a view to specify ψ_1, ψ_2:

$$\text{(4.1)} \qquad \begin{cases} \dfrac{\omega_1^{(k+1)} - \omega_1^{(k)}}{\tau} = \dfrac{\partial}{\partial x}\left(\dfrac{\partial\omega_1^{(k)}}{\partial x} + \dfrac{\partial\omega_2^{(k)}}{\partial y} - \dfrac{z}{\alpha}\right) - C\omega_1^{(k+1)}, \\ \dfrac{\omega_2^{(k+1)} - \omega_2^{(k)}}{\tau} = \dfrac{\partial}{\partial y}\left(\dfrac{\partial\omega_1^{(k)}}{\partial x} + \dfrac{\partial\omega_2^{(k)}}{\partial y} - \dfrac{z}{\alpha}\right) - C\omega_2^{(k+1)}, \end{cases}$$

where $C = \sqrt{\left(\frac{\partial}{\partial x}(\frac{\partial\omega_1^{(k)}}{\partial x} + \frac{\partial\omega_2^{(k)}}{\partial y} - \frac{z}{\alpha})\right)^2 + \left(\frac{\partial}{\partial y}(\frac{\partial\omega_1^{(k)}}{\partial x} + \frac{\partial\omega_2^{(k)}}{\partial y} - \frac{z}{\alpha})\right)^2}$ will be 'frozen' locally to implement the Fourier analysis [2]. The finite difference discretization takes the form

$$\text{(4.2)} \begin{cases} \dfrac{(\omega_1^{(k+1)})_{i,j} - (\omega_1^{(k)})_{i,j}}{\tau} = \dfrac{(\omega_1^{(k)})_{i+1,j} + (\omega_1^{(k)})_{i-1,j} - 2(\omega_1^{(k)})_{i,j}}{h^2} \\ \qquad + \dfrac{(\omega_2^{(k)})_{i+1,j} + (\omega_2^{(k)})_{i,j-1} - (\omega_2^{(k)})_{i,j} - (\omega_2^{(k)})_{i+1,j-1}}{h^2} \\ \qquad - \dfrac{D_x^+ z}{\alpha} - C\omega_1^{(k+1)}, \\ \dfrac{(\omega_2^{(k+1)})_{i,j} - (\omega_2^{(k)})_{i,j}}{\tau} = \dfrac{(\omega_2^{(k)})_{i,j+1} + (\omega_2^{(k)})_{i,j-1} - 2(\omega_2^{(k)})_{i,j}}{h^2} \\ \qquad + \dfrac{(\omega_1^{(k)})_{i+1,j} + (\omega_1^{(k)})_{i,j-1} - (\omega_1^{(k)})_{i,j} - (\omega_1^{(k)})_{i+1,j-1}}{h^2} \\ \qquad - \dfrac{D_y^+ z}{\alpha} - C\omega_2^{(k+1)}. \end{cases}$$

Applying the Fourier analysis with $h = 1, n = 16$ on a $n \times n$ grid to (4.2) leads to

$$\begin{cases} (e_1^{(k+1)})_{i,j}(1 + \tau C) = (e_1^{(k)})_{i,j}(1 - 2\tau) + \tau\left[(e_1^{(k)})_{i+1,j} + (e_1^{(k)})_{i-1,j}\right] + \\ \qquad \tau\left[(e_2^{(k)})_{i+1,j} + (e_2^{(k)})_{i,j-1} - (e_2^{(k)})_{i+1,j-1} - (e_2^{(k)})_{i,j}\right], \\ (e_2^{(k+1)})_{i,j}(1 + \tau C) = (e_2^{(k)})_{i,j}(1 - 2\tau) + \tau\left[(e_2^{(k)})_{i,j+1} + (e_2^{(k)})_{i,j-1}\right] + \\ \qquad \tau\left[(e_1^{(k)})_{i-1,j} + (e_1^{(k)})_{i,j+1} - (e_1^{(k)})_{i-1,j+1} - (e_1^{(k)})_{i,j}\right], \end{cases}$$

and further to

$$\text{(4.3)}\ \mathcal{A}_{\alpha,\beta} = \begin{bmatrix} \frac{2\,\tau\,\cos(\pi\,\alpha/8)+1-2\,\tau}{1+C\tau} & \frac{\tau\left(e^{i\pi\,\alpha/8}+e^{-\,i\pi\,\beta/8}-1-e^{i\pi\,(\alpha-\beta)/8}\right)}{1+C\tau} \\ \frac{\tau\left(e^{-i\pi\,\alpha}/8+e^{i\pi\,\beta/8}-1-e^{-\,i\pi\,(\alpha-\beta)/8}\right)}{1+C\tau} & \frac{2\,\tau\,\cos(\pi\,\beta/8)+1-2\,\tau}{1+C\tau} \end{bmatrix}.$$

ETNA
Kent State University
etna@mcs.kent.edu

TABLE 4.1
Estimated smoothing rate of the Chambolle's method

Minimal C value	$\max_{\alpha,\beta} \rho(\mathcal{A}_{\alpha,\beta})$	Predicted iterations for 10^{-6}
10	0.4444	17
1	0.8889	117
1/2	0.9412	228
1/4	0.9697	449
10^{-1}	0.9877	1116
10^{-2}	0.9988	11506
10^{-3}	0.9999	138150

Then to compute the smoothing rate $\max_{\alpha,\beta} \rho(\mathcal{A}_{\alpha,\beta})$ in the high frequency range $(\alpha, \beta) \in [-\pi, \pi]\backslash[-\pi/2, \pi/2]$, we have to estimate this quantity numerically on same mesh grid and each specification of C; with $\tau = 1/8$ as in [5] we found that the convergence rate is identical to the smoothing rate shown in Table 4.1, where the last column shows the predicted number of iterations to reduce a residual by 10^{-6}. Other numerical tests, based on using exact values for $\boldsymbol{\omega}, C$ at pixels where C is small, confirmed the sharpness of this last column. The results confirm the belief that the behaviour of the dual equation is not similar to what is expected of an elliptic equation. This analysis also shows that when there are a substantial number of pixels at which $C = |\nabla u| < 1/100$, the smoothing rate of 0.9988 will be too close to 1, i.e., the Chambolle iterations fail to provide an effective smoother. This case will happen in flat regions of the desired image u. Since such a quantity will gradually get small in such flat regions as iterations progress, one should observe that, correspondingly, the multigrid algorithm should become slower and slower. Indeed this phenomenon has been observed.

It remains to understand more of the nature of slow convergence of the Chambolle's iterations associated with small C and to propose new methods to speed up the iterations.

4.2. Analysis for a modified smoother. A simple generalization of Chambolle smoothing iterations is to introduce a parameter $s \in \mathbb{R}$ into the basic scheme

$$\left\{ \begin{aligned} \frac{\omega_1^{(k+1)} - \omega_1^{(k)}}{\tau} &= \frac{\partial}{\partial x}\left(\frac{\partial \omega_1^{(k)}}{\partial x} + \frac{\partial \omega_2^{(k)}}{\partial y} - \frac{z}{\alpha}\right) \\ &\qquad - s\omega_1^{(k+1)} - C\omega_1^{(k+1)} + s\omega_1^{(k)} + f_1, \\ \frac{\omega_2^{(k+1)} - \omega_2^{(k)}}{\tau} &= \frac{\partial}{\partial y}\left(\frac{\partial \omega_1^{(k)}}{\partial x} + \frac{\partial \omega_2^{(k)}}{\partial y} - \frac{z}{\alpha}\right) \\ &\qquad - s\omega_2^{(k+1)} - C\omega_2^{(k+1)} + s\omega_2^{(k)} + f_2, \end{aligned} \right. \tag{4.4}$$

where $s = f_1 = f_2 = 0$ reduce to (2.4) on the finest level, f_1, f_2 are normally not 0 on coarse levels and

$$C = \sqrt{\Big(\frac{\partial}{\partial x}(\frac{\partial \omega_1^{(k)}}{\partial x} + \frac{\partial \omega_2^{(k)}}{\partial y} - \frac{z}{\alpha})\Big)^2 + \Big(\frac{\partial}{\partial y}(\frac{\partial \omega_1^{(k)}}{\partial x} + \frac{\partial \omega_2^{(k)}}{\partial y} - \frac{z}{\alpha})\Big)^2}.$$

ETNA
Kent State University
etna@mcs.kent.edu

TABLE 4.2
Estimated smoothing rate of a modified Chambolle type method with $s = -4$

Minimal C value	$\max_{\alpha,\beta} \rho(\mathcal{A}_{\alpha,\beta})$	Predicted iterations for 10^{-6}
10	0.2857	11
1	0.8000	62
1/2	0.8889	117
1/4	0.9412	228
10^{-1}	0.9756	559
10^{-2}	0.9980	6901
10^{-3}	0.9998	69071

The finite difference discretization similar to (4.2) takes the form

$$
(4.5)\quad \begin{cases}
\dfrac{(\omega_1^{(k+1)})_{i,j} - (\omega_1^{(k)})_{i,j}}{\tau} = \dfrac{(\omega_1^{(k)})_{i+1,j} + (\omega_1^{(k)})_{i-1,j} - 2(\omega_1^{(k)})_{i,j}}{h^2} \\
\qquad + \dfrac{(\omega_2^{(k)})_{i+1,j} + (\omega_2^{(k)})_{i,j-1} - (\omega_2^{(k)})_{i,j} - (\omega_2^{(k)})_{i+1,j-1}}{h^2} \\
\qquad - \dfrac{D_x^+ z}{\alpha} - s\omega_1^{(k+1)} - C\omega_1^{(k+1)} + s\omega_1^{(k)} + f_1, \\
\dfrac{(\omega_2^{(k+1)})_{i,j} - (\omega_2^{(k)})_{i,j}}{\tau} = \dfrac{(\omega_2^{(k)})_{i,j+1} + (\omega_2^{(k)})_{i,j-1} - 2(\omega_2^{(k)})_{i,j}}{h^2} \\
\qquad + \dfrac{(\omega_1^{(k)})_{i+1,j} + (\omega_1^{(k)})_{i,j-1} - (\omega_1^{(k)})_{i,j} - (\omega_1^{(k)})_{i+1,j-1}}{h^2} \\
\qquad - \dfrac{D_y^+ z}{\alpha} - s\omega_2^{(k+1)} - C\omega_2^{(k+1)} + s\omega_2^{(k)} + f_2.
\end{cases}
$$

Applying the Fourier analysis to (4.5) leads to the matrix

$$
\mathcal{A}_{\alpha,\beta} = \begin{bmatrix} \frac{2\,\tau\,\cos(\pi\,\alpha/8)+1+s\tau-2\,\tau}{1+C\tau+s\tau} & \frac{\tau\left(e^{i\pi\,\alpha/8}+e^{-i\pi\,\beta/8}-1-e^{i\pi\,(\alpha-\beta)/8}\right)}{1+C\tau+s\tau} \\ \frac{\tau\left(e^{-i\pi\,\alpha/8}+e^{i\pi\,\beta/8}-1-e^{-i\pi\,(\alpha-\beta)/8}\right)}{1+C\tau+s\tau} & \frac{2\,\tau\,\cos(\pi\,\beta/8)+1+s\tau-2\,\tau}{1+C\tau+s\tau} \end{bmatrix}.
$$

Then estimating $\max_{\alpha,\beta} \rho(\mathcal{A}_{\alpha,\beta})$ in the high frequency range $(\alpha, \beta) \in [-\pi, \pi] \backslash [-\pi/2, \pi/2]$ numerically gives the results in Table 4.2, where $s = -4$ gives much improved rates than the standard Chambolle's iterations. Intuitively one might suggest that $s = 2$ leads to a good smoother but this is not the case.

In experiments, the use of $s = -4$ can indeed accelerate both the Chambolle iterations and the corresponding multigrid method by up to 50%.

4.3. Modified algorithms. In designing modified algorithms, the main consideration is how to overcome the slow convergence of the Chambolle type methods, either (4.1) or (4.4), for small C.

4.3.1. A simple multigrid algorithm. Having analyzed the convergence and smoothing rates of the Chambolle's iterations, a simpler idea is to ensure that the C term is never too small, so that multigrid algorithms would converge quickly.

We reconsider the modified system (3.2)

$$
\frac{\boldsymbol{\omega}_{i,j}^{(k+1)} - \boldsymbol{\omega}_{i,j}^{(k)}}{\tau} = \left(\nabla(\mathrm{div}(\boldsymbol{\omega}^{(k)}) - \frac{z}{\alpha})\right)_{i,j} - \sqrt{\left|\nabla(\mathrm{div}(\boldsymbol{\omega}^{(k)}) - \frac{z}{\alpha})\right|_{i,j}^2 + \beta}\; \boldsymbol{\omega}_{i,j}^{(k+1)}
$$

ETNA
Kent State University
etna@mcs.kent.edu

and in particular (with $s = -4$)

$$\frac{\boldsymbol{\omega}_{i,j}^{(k+1)} - \boldsymbol{\omega}_{i,j}^{(k)}}{\tau} = \left(\nabla(\mathrm{div}(\boldsymbol{\omega}^{(k)}) - \frac{z}{\alpha})\right)_{i,j} - C_\beta \boldsymbol{\omega}_{i,j}^{(k+1)} - s\boldsymbol{\omega}_{i,j}^{(k+1)} + s\boldsymbol{\omega}_{i,j}^{(k)}, \tag{4.6}$$

where $\beta = 10^{-2}$ instead of $\beta = 10^{-3}$ tried previously.

With this choice, $C_\beta \geq 10^{-2}$ so the smoothing rate is $\rho \leq 0.998$ and 360 steps would reduce a residual by 0.4962 which resembles the smoothing rate by a Gauss-Seidel method for a Poisson's equation [2]. In practice, pre-smoothing steps ranging in $4 - 40$ would be sufficient to generate a converging multigrid method.

4.3.2. A semi-implicit method. A natural idea to increase the 'implicitness' of the previous scheme (2.5) is by

$$\frac{\boldsymbol{\omega}_{i,j}^{(k+1)} - \boldsymbol{\omega}_{i,j}^{(k)}}{\tau} = \left(\nabla(\mathrm{div}(\boldsymbol{\omega}^{(k+1)}) - \frac{z}{\alpha})\right)_{i,j} - \left|\nabla(\mathrm{div}(\boldsymbol{\omega}^{(k)}) - \frac{z}{\alpha})\right|_{i,j} \boldsymbol{\omega}_{i,j}^{(k+1)}, \tag{4.7}$$

which leads to no improvements over (2.5). To understand the problem with $C \approx 0$, recognizing that $C = |\nabla u|$, it is constructive to consider the primal-dual system (2.7)

$$\begin{cases} \alpha \mathrm{div}(\boldsymbol{\omega}) + u - z = 0, \\ \nabla u + C\boldsymbol{\omega} = 0, \end{cases} \quad \text{or} \quad \begin{cases} \alpha \mathrm{div}(\boldsymbol{\omega}^{(k+1)}) + u^{(k+1)} - z = 0, \\ \nabla u^{(k+1)} + C^{(k)}\boldsymbol{\omega}^{(k+1)} = 0. \end{cases}$$

Put into matrix form on discretization, the above system can be written as

$$\begin{bmatrix} I/\alpha & P & Q \\ U & C_1 & 0 \\ V & 0 & C_2 \end{bmatrix} \begin{bmatrix} u \\ \omega_1 \\ \omega_2 \end{bmatrix}^{(k+1)} = \begin{bmatrix} z/\alpha \\ 0 \\ 0 \end{bmatrix}, \tag{4.8}$$

where C_1, C_2, each of size $(n-1)n \times (n-1)n$, denote two diagonal matrices consisted of entries of $C^{(k)}, P, Q$ denote the divergence operator div and U, V denote the gradient operator ∇ for vector forms of u, z, of size $n^2 \times 1$. Clearly if there are sufficiently large number of extremely small diagonal (or 0) entries in C_1, C_2 due to C, the coefficient matrix of the linearized primal-dual system (4.8) will be singular!

One way to ensure that (4.8) is non-singular is to add a positive parameter γ on the main diagonal, so it becomes

$$\begin{bmatrix} I/\alpha & P & Q \\ U & C_1 + \gamma I & 0 \\ V & 0 & C_2 \end{bmatrix} \begin{bmatrix} u \\ \omega_1 \\ \omega_2 \end{bmatrix}^{(k+1)} = \begin{bmatrix} z/\alpha \\ \gamma\omega_1^{(k)} \\ 0 \end{bmatrix}, \tag{4.9}$$

which will have the same solution as (4.8) upon convergence. This purely algebraic consideration is in fact equivalent to solving the partially time-marching primal-dual system

$$\begin{cases} \alpha \mathrm{div}(\boldsymbol{\omega}^{(k+1)}) + u^{(k+1)} &= z, \\ \dfrac{\omega_1^{(k+1)} - \omega_1^{(k)}}{\tau} &= -\dfrac{\partial u^{(k+1)}}{\partial x} - C^{(k)}\omega_1^{(k+1)}, \\ \dfrac{\partial u^{(k+1)}}{\partial y} + C^{(k)}\omega_2^{(k+1)} &= 0, \end{cases}$$

if one chooses $\tau\gamma = 1$.

TABLE 5.1
Comparison of a nonlinear multilevel algorithm to the Chambolle's method with tol=10^{-4}

Problem number	Size n	Chambolle Steps	CPU	PNSR	Multigrid Cycles	CPU	PNSR
1	63	55675	325.7	24.81	4	2.5	24.88
	128	121926	2473.8	26.94	4	9.1	27.06
	256	212544	22978.8	29.71	4	48.8	29.88
2	63	38529	226.3	25.53	4	2.9	25.63
	128	82858	1675.4	27.60	5	9.7	27.64
	256	212887	20309.0	29.98	5	43.2	29.95

TABLE 5.2
Performance of a nonlinear multilevel algorithm with tol=10^{-4}

Problem number	Size n	Multigrid Cycles	CPU	PNSR
1	511	5	158.9	32.90
	1023	5	1277.6	35.75
	2047	6	8365.2	36.38
2	511	5	163.0	32.43
	1023	5	1284.3	34.55
	2047	6	8191.6	36.78

5. Numerical experiments. In this section, we shall first compare Algorithm 3.1 using the smoother (4.6) with the Chambolle's iterative method (2.5) in both solution quality and speed of convergence. Then we demonstrate the performance of Algorithm 3.1 for some large images which cannot be processed by the Chambolle's iterative method (2.5) in an acceptable amount of time. Finally we present some preliminary results from using the linearized primal-dual iterations (4.9) as an iterative method, whose effective use in a multilevel algorithm will be the subject of future work.

We shall use the two examples shown in Figs. 3.1-3.2 in our experiments. The stopping criterion will be based on reducing the static residual, i.e., (2.4), to below a tolerance tol. Restoration performance is quantitatively measured by the peak signal-to-noise ratio (PSNR)

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

where $r_{i,j}$ and $u_{i,j}$ denote the pixel values of the restored image and the original image respectively, with $u, r \in \mathbb{R}^{m\times n}$. Here we assume $r_{i,j}, u_{i,j} \in [0, 255]$.

Comparison. We take several resolutions of the same problems and display the results in Table 5.1. Clearly one observes that the multigrid method is much more efficient than the Chambolle's method for delivering the same quality (PSNR) of restoration.

Large images. From the values of CPU in Table 5.1, one may envisage that getting the results for $n \geq 511$ by Chambolle's method can take many days, especially for small *tol* (we should remark that there are practical situations where restored results are already acceptable before convergence). Instead, we test the examples with large n for the multigrid method and show the results in Table 5.2. Clearly one sees that the efficiency scales well with n. The restored $n \times n$ images for $n = 1023$ are also shown in Fig. 5.1.

Performance of linearized primal-dual iterations. Finally we test the linearized primal-dual method (LPD) (4.9) and show some results in Table 5.3. For the same tol=10^{-4}, we see that

ETNA
Kent State University
etna@mcs.kent.edu

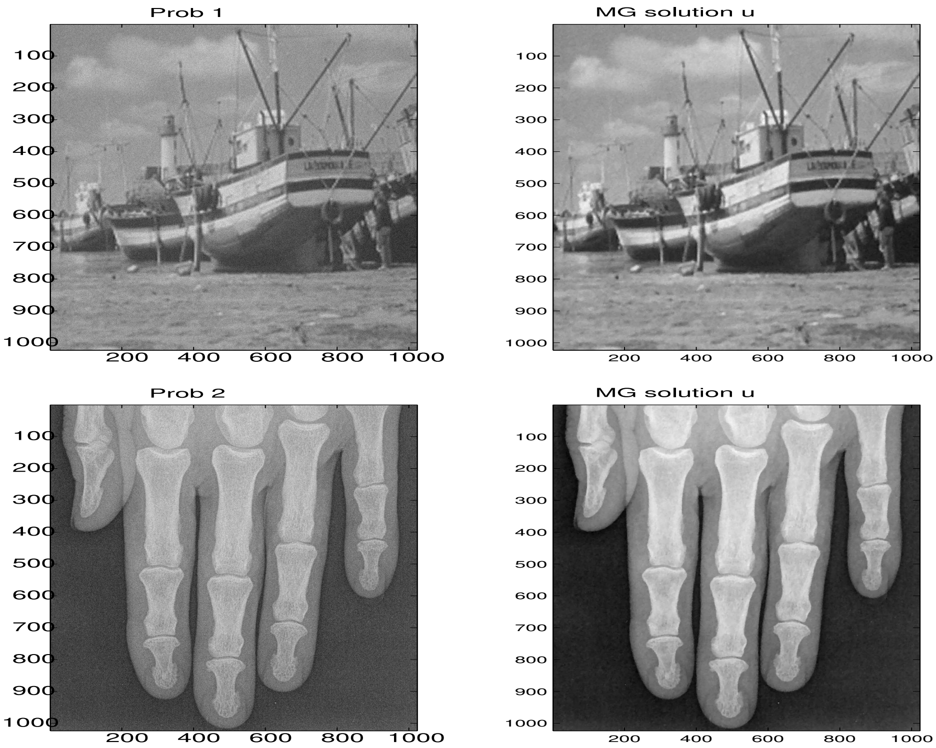

FIG. 5.1. *Restored results from a multigrid method, with the left and the right plots displaying respectively the noisy image and the restored image by* 4 *cycles of MG. Here* $\beta = 10^{-2}$ *for both problems. For problem* 1*:* $\alpha = 20, psnr(z) = 22.1, psnr(u_{MG}) = 33.9$ *and* $cpu(u_{MG}) = 1151.95$*. For problem* 2*:* $\alpha = 40, psnr(z) = 22.1, psnr(u_{MG}) = 30.5$ *and* $cpu(u_{MG}) = 1158.15$.

TABLE 5.3
Performance of a linearized primal-dual algorithm with tol$=10^{-4}$

Problem number	Size n	LPD Steps	CPU	PNSR
1	63	255	198.0	24.81
	127	348	1481.9	26.94
	255	432	11040.4	29.71
2	63	261	191.2	25.53
	127	340	1379.6	27.60
	255	461	11300.6	29.98

LPD takes much less iterations then the Chambolle's method; see Table 5.1.

6. Conclusions. This paper presented two iterative solvers for solving the dual formulation for image restoration. After giving a unified formulation of the dual problem, we analyzed the convergence rate of the widely-used Chambolle's method and found that $C = |\nabla u| \approx 0$ contributes to the slow convergence of the method. We then proposed an improved method suitable for use in multigrid methods to solve a regularized version of the dual formulation. We also considered a linearized primal-dual iteration method for the dual problem. Numerical results confirmed that the multigrid method with a modified Chambolle's smoother is many orders of magnitude faster than the original method. The linearized primal-dual iteration method is found to give fast convergence, but it has yet to be used in a multigrid method.

REFERENCES

[1] G. AUBERT AND P. KORNPROBST, *Mathematical Problems in Image Processing: Partial Differential Equations and the Calculus of Variations*, Springer, Berlin, 2001.

[2] A. BRANDT, *Multilevel adaptive solutions to boundary value problems*, Math. Comp., 31 (1977), pp. 333–390.

[3] A. BRANDT, *Relaxation methods for the dual formulation*, private communication, 2006.

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

[5] A. CHAMBOLLE, *An algorithm for total variation minimization and applications*, J. Math. Imaging Vision, 20 (2004), pp. 89–97.

[6] T. F. CHAN AND K. CHEN, *On a nonlinear multigrid algorithm with primal relaxation for the image total variation minimisation*, Numer. Algorithms, 41 (2006), pp. 387–411.

[7] T. F. CHAN AND K. CHEN, *An optimization-based multilevel algorithm for total variation image denoising*, Multiscale Model. Simul., 5 (2006), pp. 615–645.

[8] T. F. CHAN, K. CHEN, AND X. C. TAI, *Nonlinear multilevel schemes for solving the total variation image minimization problem*, in Image Processing Based on PDEs, X.-C. Tai, K. - A. Lie, T. F. Chan, and S. Osher, eds., Springer, Berlin, 2006, pp .265–288.

[9] T. F. CHAN, S. ESEDOGLU, AND F. E. PARK, *A fourth order dual method for staircase reduction in texture extraction and image restoration problems*, UCLA CAM report 05-28, April 2005.

[10] 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 (1999), pp. 1964–1977.

[11] T. F. CHAN AND J. H. SHEN, *Image Processing And Analysis: Variational, PDE, Wavelet, and Stochastic Methods*, SIAM, Phildelphia, 2005.

[12] K. CHEN, *Matrix Preconditioning Techniques and Applications*, Cambridge Monographs on Applied and Computational Mathematics, No. 19, Cambridge University Press, UK, 2005.

[13] C. GROETSCH AND O. SCHERZER, *Nonstationary iterated Tikhonov-Morozov method and third order differential equations for the evaluation of unbounded operators*, Math. Methods Appl. Sci., 23 (2000), pp. 1287–1300.

[14] M. HINTERMÜLLER AND G. STADLER, *An infeasible primal-dual algorithm for total bounded variation-based inf-convolution-type image restoration*, SIAM. J. Sci. Comput., 28 (2006), pp. 1–23.

[15] Y. MEYER, *Oscillating Patterns in Image Processing and Nonlinear Evolution Equations*, University Lecture Series, Vol. 22, American Mathematical Society, Providence, RI, 2001.

[16] M. NG, L. Q. QI, Y. F. YANG, AND Y. M. HUANG, *On semismooth Newton's methods for total variation minimization*, J. Math. Imaging Vision, 27 (2007), pp. 265–276.

[17] P. PERONA AND J. MALIK, *Scale-space and edge detection using anisotropic diffusion*, IEEE TPAMI, 12 (1990), pp. 629–639.

[18] L. I. RUDIN, S. OSHER, AND E. FATEMI, *Nonlinear total variation based noise removal algorithms*, Physica D, 60 (1992), pp. 259–268.

[19] J. SAVAGE AND K. CHEN, *On multigrids for solving a class of improved total variation based PDE models*, in Image Processing Based On Partial Differential Equations, X.-C. Tai, K.-A. Lie, T. F. Chan, and S. Osher, eds., Springer, Berin, 2006, pp. 69–94.

[20] U. TROTTENBERG, C. OOSTERLEE, AND A. SCHULLER, *Multigrid*, Academic Press, London, 2001.