An Improved and Accelerated Nonlinear Multigrid Method for Total-Variation Denoising

Joseph Savage* and Ke Chen†

Department of Mathematical Sciences,
The University of Liverpool,
Peach Street, Liverpool L69 7ZL, UK.

©IJCM (2005) Vol 82 (8), pp.1001-1015

Abstract

Fast solution of the nonlinear partial differential equations (PDEs) arising from image restoration is of practical importance. The standard multigrid methods do not work well, due to highly discontinuous coefficients of the underlying nonlinear PDEs. In this paper, we present two related global but linear smoothers that help the convergence of multigrid methods. Furthermore, the Krylov acceleration technique is combined with the proposed multigrid method to improve the performance. Numerical experiments are shown.

Keywords. Image restoration, total variation, denoising, multigrid methods, Krylov acceleration, smoothers.

1 Introduction

During recording and transmission an image will become contaminated with random noise; this is modelled by the equation

$$z(x,y) = u(x,y) + n(x,y), \qquad x,y \in \Omega \tag{1}$$

where Ω is a bounded and open domain of $\mathbb{R}^2$ (usually a rectangle). Here z is a real function representing the observed image, which in practice will only be known at certain discrete values of x and y, u is the true image and n is an additive noise term. The recovery of the original image from the noisy observed one is known as denoising.

The aim of a denoising process is to remove noise while preserving the main features of the original image, most importantly its edges (an edge is a boundary where a jump in intensity occurs); this can be achieved by the use of Tikhonov regularization with the Total-Variation (TV) regularization functional, first introduced in 1992 by Rudin Osher and Fatemi [7]. Tikhonov regularization involves the minimization of a penalized least squares functional

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

where R is a regularization functional that penalizes against certain artifacts in the solution and α is a regularization parameter that balances the need for a solution which is a good fit to the observed data and one that is regular. Traditional regularization functionals penalized against non-smooth solutions and therefore had the unfortunate effect of smudging edges in the original image, the TV regularization

*Support of the UK EPSRC DTA grant is gratefully received.
†E-mail: k.chen@liverpool.ac.uk, URL: http://www.liv.ac.uk/∼cmchenke (for correspondence).

functional on the other hand allows solutions which are piecewise smooth and can therefore preserve edges, it is defined as follows

$$TV(u) = \int_{\Omega} |\nabla u| dxdy. \tag{3}$$

Tikhonov regularization with this regularization functional results in the Euler-Lagrange equation

$$u - \alpha \nabla . \left(\frac{\nabla u}{|\nabla u|} \right) = z. \tag{4}$$

This equation is not well defined when $|\nabla u| = 0$ and so the TV functional is perturbed slightly by a small positive parameter β

$$TV_{\beta}(u) = \int_{\Omega} \sqrt{|\nabla u|^2 + \beta} dxdy \tag{5}$$

which results in the Euler-Lagrange equation

$$u - \alpha \nabla . \left(\frac{\nabla u}{\sqrt{|\nabla u|^2 + \beta}} \right) = z \tag{6}$$

with homogenous Neumann boundary condition $\frac{\partial u}{\partial n} = 0$ at the boundary.

There are several iterative methods for solving (6). In [9], Vogel and Oman use a "lagged diffusivity fixed point iteration" in which a linear equation

$$u^{k+1} - \alpha \nabla . \left(\frac{\nabla u^{k+1}}{\sqrt{|\nabla u^k|^2 + \beta}} \right) = z \tag{7}$$

is solved at each step to obtain the new iterate u^{k+1} from the current iterate u^k. The matrix (which is dependant on u^k) resulting from discretization of this equation is SPD. They use a preconditioned conjugate gradient method with a multigrid preconditioner to solve the linear system at each step.

Newton's Method has a small domain of convergence for this problem, particularly for small values of the parameter β, and a continuation procedure on this parameter is required [4]. To overcome this, in [3], Chan Golub and Mulet use a Primal-Dual Newton Method to solve (6). The Euler-Lagrange equation is replaced by the equivalent system

$$\begin{cases} -\alpha \nabla . w + u - z = 0 \\ \\ w\sqrt{\|\nabla u\|^2 + \beta} - \nabla u = 0 \end{cases} \tag{8}$$

where

$$w = \frac{\nabla u}{\sqrt{\|\nabla u\|^2 + \beta}}. \tag{9}$$

The (u, w) system is linearized by an approximate Newton's method and the inner linear iterations are done using a preconditioned conjugate gradient method.

In [7], Rudin, Osher and Fatemi use an explicit time marching method to find the steady state of the parabolic equation

$$\frac{\partial u}{\partial t} = (u - z) - \alpha \nabla . \left(\frac{\nabla u}{\sqrt{|\nabla u|^2 + \beta}} \right). \tag{10}$$

More recently in [6], Marquina and Osher reduce the stability constraints on the time step for this method by multiplying the right hand side of the equation by $|\nabla u|$ in an improved scheme.

In this paper we propose a non-linear multigrid method based on the full approximation scheme for solving (6) using as a smoother the fixed point method with just a few steps of specially adapted Gauss-Seidel relaxation applied to the linear equation at each step. In section 2 we review the usual discretization and multigrid method and in section 3 we discuss our suggested smoothers for the nonlinear multigrid algorithm with some Krylov acceleration technique considered in section 4. Finally in section 5 various numerical results are presented.

2 A nonlinear multigrid method

In our work we assume Ω is a rectangular domain $[a,b]\times[c,d]$ and we discretize it in a cell centered fashion, suitable for the finite difference method. If the observed image z is in the form of $n\times m$ values each representing the light intensity at a pixel, then we split the domain into $n\times m$ cells of size $h\times k$ and place a grid point at the centre of each cell where $h=(b-a)/n$ is the grid spacing in the x-direction and $k=(d-c)/m$ is the grid spacing in the y-direction. The grid point (i,j) is located at

$$(x_i,y_j)=\left(a+\frac{(2i-1)h}{2},c+\frac{(2j-1)k}{2}\right),\qquad 1\le i\le n,1\le j\le m. \tag{11}$$

The value of z at grid point (i,j) is denoted by $z_{i,j}$. The Euler-Lagrange equation (6) is discretized using a finite difference scheme. The equation at grid point (i,j) is

$$u_{i,j}-\alpha\left[\frac{\delta_x^-}{h}\left(\frac{\delta_x^+u_{i,j}/h}{\sqrt{(\delta_x^+u_{i,j}/h)^2+(\delta_y^+u_{i,j}/k)^2+\beta}}\right)+\frac{\delta_y^-}{k}\left(\frac{\delta_y^+u_{i,j}/k}{\sqrt{(\delta_x^+u_{i,j}/h)^2+(\delta_y^+u_{i,j}/k)^2+\beta}}\right)\right]=z_{i,j}, \tag{12}$$

where

$$\delta_x^\pm u_{i,j}=\pm\left(u_{i\pm1,j}-u_{i,j}\right),\qquad \delta_y^\pm u_{i,j}=\pm\left(u_{i,j\pm1}-u_{i,j}\right), \tag{13}$$

which can be rewritten as

$$u_{i,j}-\alpha_h\left[\delta_x^-\left(\frac{\delta_x^+u_{i,j}}{\sqrt{(\delta_x^+u_{i,j})^2+(\lambda\delta_y^+u_{ij})^2+\beta_h}}\right)+\lambda\delta_y^-\left(\frac{\lambda\delta_y^+u_{i,j}}{\sqrt{(\delta_x^+u_{i,j})^2+(\lambda\delta_y^+u_{ij})^2+\beta_h}}\right)\right]=z_{ij} \tag{14}$$

where

$$\alpha_h=\alpha/h,\ \beta_h=h^2\beta \text{ and } \lambda=h/k \tag{15}$$

with boundary condition

$$u_{i,0}=u_{i,1},u_{i,m+1}=u_{i,m},u_{0,j}=u_{1,j},u_{n+1,j}=u_{n,j}. \tag{16}$$

The full approximation scheme (multigrid method). Denote by

$$N^hu^h=z^h \tag{17}$$

the nonlinear system of equations described by (14)-(16) where u^h and z^h are grid functions on an $n\times m$ cell-centered rectangular grid Ω^h with grid spacing (h,k). Denote by Ω^{2h} the $n/2\times m/2$ cell-centered grid which results from standard coarsening of Ω^h i.e the cell-centered grid with grid spacing $(2h,2k)$. If

v^h is an approximation to the solution of (17) define the error in v^h by $e^h = u^h - v^h$ and the residual by $r^h = z^h - N^h v^h$ recall also that these quantities are related by the nonlinear residual equation:

$$N^h(v^h + e^h) - N^h v^h = r^h \tag{18}$$

If e^h is smooth it can be well approximated on Ω^{2h}. Any iterative method which smooths the error on the fine grid can therefore be improved by the use of coarse grid correction, in which a coarse grid analogue of the residual equation is solved (solution on the coarse grid being less expensive than on the fine grid) to obtain a coarse grid approximation of the error, which is then transferred back to the fine grid to correct the approximation v^h. This is known as a two-grid cycle and with recursive application can be extended to a multigrid method. Below we define restriction and interpolation operators for transferring grid functions between Ω^h and Ω^{2h} before introducing the smoothing method we use for our multigrid method.

Restriction.

$$I_h^{2h} v^h = v^{2h} \tag{19}$$

where

$$v_{i,j}^{2h} = \frac{1}{4}(v_{2i-1,2j-1}^h + v_{2i-1,2j}^h + v_{2i,2j-1}^h + v_{2i,2j}^h) \qquad 1 \le i \le n/2, 1 \le j \le m/2 \tag{20}$$

Interpolation.

$$I_{2h}^h v^{2h} = v^h \tag{21}$$

where

$$\begin{aligned}
v_{2i,2j}^h &= \tfrac{1}{16}(9v_{i,j}^{2h} + 3(v_{i+1,j}^{2h} + v_{i,j+1}^{2h}) + v_{i+1,j+1}^2 h) \\
v_{2i-1,2j}^h &= \tfrac{1}{16}(9v_{i,j}^{2h} + 3(v_{i-1,j}^{2h} + v_{i,j+1}^{2h}) + v_{i-1,j+1}^2 h) \\
v_{2i,2j-1}^h &= \tfrac{1}{16}(9v_{i,j}^{2h} + 3(v_{i+1,j}^{2h} + v_{i,j-1}^{2h}) + v_{i+1,j-1}^2 h) \\
v_{2i-1,2j-1}^h &= \tfrac{1}{16}(9v_{i,j}^{2h} + 3(v_{i-1,j}^{2h} + v_{i,j-1}^{2h}) + v_{i-1,j-1}^2 h) \\
&1 \le i \le n/2, 1 \le j \le m/2
\end{aligned} \tag{22}$$

Choice of smoothers. We tried several iterative methods as smoothers for our multigrid method. Initially we tried using a Gauss-Seidel Newton method one step of which involves cycling through each grid point (i, j) in turn and substituting into (14) current values of the approximation corresponding to all grid points except (i, j) to give a nonlinear equation in one variable. Several steps of Newton method are applied to this nonlinear equation to update the approximation at (i, j). However we found that this method was divergent. This confirms the well-known fact that the standard multigrid method does not perform well for the TV equation [2].

3 Improved smoothers and the overall multigrid algorithm

We have found two less standard smoothers that work well with the standard nonlinear multigrid method [8], although the second smoother will be recommended.

Smoother S1. In order to explain this method, we rewrite (14) as

$$u_{i,j} - \alpha_h((D(u)_{i,j}(u_{i+1,j} - u_{i,j}) - D(u)_{i-1,j}(u_{i,j} - u_{i-1,j}))$$

$$+\lambda^2 \left(D(u)_{i,j}(u_{i,j+1} - u_{ij}) - D(u)_{i,j-1}(u_{i,j} - u_{i,j-1}))\right) = z_{i,j} \tag{23}$$

where

$$D(u)_{i,j} = ((\delta_x^+ u_{i,j})^2 + (\lambda \delta_y^+ u_{i,j})^2 + \beta_h)^{-1/2}. \tag{24}$$

Note that $D(u)_{ij}$, $D(u)_{i-1,j}$ and $D(u)_{i,j-1}$ all contain u_{ij} terms. Then, for each grid point (i,j) as well as substituting current values of the approximation at points other than (i,j) into (23) we also substitute current values of the approximation at (i,j) into the D terms; this gives us a linear equation in one variable to solve in order to update the approximation at (i,j). The algorithm for updating v^h using this method is given below:

$$v^h \leftarrow S1(v^h, z^h, maxit, tol)$$

for $i = 1 : n$

for $j = 1 : m$

for $iter = 1 : maxit$

$\bar{v}^h \leftarrow v^h$

update $v_{i,j}$ by solving the linear equation

$$v_{i,j} - \alpha_h((D(\bar{v})_{i,j}(\bar{v}_{i+1,j} - v_{i,j}) - D(\bar{v})_{i-1,j}(v_{i,j} - \bar{v}_{i-1,j})) +$$

$$\lambda^2 \left(D(\bar{v})_{i,j}(\bar{v}_{i,j+1} - v_{ij}) - D(\bar{v})_{i,j-1}(v_{i,j} - \bar{v}_{i,j-1}))\right) = z_{i,j}$$

if $|v_{i.j} - \bar{v}_{i,j}| < tol$ stop

end

end

end

Note that because the linear equation, used to update $v_{i,j}$, involves $\bar{v}_{i,j}$ at each grid point we perform up to $maxit$ inner iterations (stopping if the modulus of the difference between $v_{i,j}$ and $\bar{v}_{i,j}$ is less than some tolerance). This method can ensure a convergent nonlinear multigrid method; however we found that the next smoother based on a global linearization of (23) is more efficient.

Smoother S2. This smoother is similar to the lagged diffusivity fixed point method of Vogel and Oman [9]. In this method the system of nonlinear equations is linearized globally at each step by evaluating $D_{i,j}$ for all (i,j) using the current approximation, several steps of Gauss-Seidel relaxation are then applied to the resulting linear system. We found that while exactly solving the linear system at each step seems to give a method with no smoothing properties, applying just a few steps of Gauss-Seidel to the linear system results in a method that while obviously slower to converge than the fixed point method, can be used as a smoother in a nonlinear multigrid method. The algorithm is given below:

$$v^h \leftarrow S2(v^h, z^h, it)$$

for $i = 1 : n$

for $j = 1 : m$

evaluate

$D(v^h)_{ij} = ((\delta_x^+ v_{i,j})^2 + (\lambda \delta_y^+ v_{i,j})^2 + \beta)^{-1/2}$

end

end

Perform Gauss-Seidel steps on linear system

$w^h = v^h$

for $iter = 1 : it$

for $i = 1 : n$

for $j = 1 : m$

$\bar{w} \leftarrow w$

$$w_{i,j} \leftarrow \frac{z_{i,j} + \alpha_h(D(v^h)_{i,j}(\bar{w}_{i+1,j} + \lambda^2\bar{w}_{i,j+1} + D(v^h)_{i-1,j}\bar{w}_{i-1,j} + \lambda^2 D(v^h)_{i,j-1}\bar{w}_{i,j-1})}{1 + \alpha_h((1+\lambda^2)D(v^h)_{i,j} + D(v^h)_{i-1,j} + \lambda^2 D(v^h)_{i,j-1})}$$

end

end

end

$v^h \leftarrow w^h$

The two-grid algorithm. We now define a two-grid method to facilitate the presentation for our nonlinear multigrid method.

Algorithm 1 (Two-grid)

(1) Apply ν_1 steps of smoothing method to $N^h u^h = z^h$ with initial guess v^h

$$v^h \leftarrow S2^{\nu_1}(v^h, z^h, it)$$

(2) Compute residual

$$r^h = z^h - N^h v^h$$

(3) Restrict residual and approximation

$$r^{2h} = I_h^{2h} r^h$$
$$v^{2h} = I_h^{2h} v^h$$

(4) Solve

$$N^{2h} w^{2h} = r^{2h} + N^{2h} v^{2h}$$

(5) Compute coarse grid error

$$e^{2h} = w^{2h} - v^{2h}$$

(6) Interpolate error

$$e^h = I_{2h}^h e^{2h}$$

(7) Correct fine grid approximation

$$v^h \leftarrow v^h + e^h$$

(8) Apply ν_2 steps of smoothing method to $N^h u^h = z^h$ with initial guess v^h

$$v^h \leftarrow S2^{\nu_2}(v^h, z^h, it).$$

Here N^{2h} is the coarse grid analogue of N^h i.e the operator resulting from discretization of the Euler-Lagrange equation on Ω^{2h}, in other words the equation at a grid point of Ω^{2h} is (14) with α_h replaced by $\alpha_{2h} = \alpha_h/2$ and β_h replaced by $\beta_{2h} = 4\beta_h$.

The multigrid algorithm. A multigrid V-cycle is obtained by replacing step (4) in Algorithm 1 with a recursive application of this coarse grid correction procedure, with the exact solution of a residual

equation taking place on some coarsest grid Ω^{Lh} (L is a power of 2). To solve on the coarsest grid Ω^{Lh} we use the primal-dual Newton method [3]. A recursive definition of the V-cycle is given below:

$$v^{h_f} \leftarrow NLMV^{h_f}(v^{h_f}, N^{h_f}, z^{h_f})$$

where h_f denotes the mesh size for the finest grid Ω^{h_f}.

Algorithm 2 (Nonlinear multigrid) *The one-step V-cycle method is defined for* $v^h \leftarrow NLMV^h(v^h, N^h, z^h)$*:*

(1) If $\Omega^h =$*coarsest grid, solve* $N^h u^h = z^h$ *using primal-dual Newton method and stop.*
Else $v^h \leftarrow S2^{\nu_1}(v^h, z^h, it)$

(2) $v^{2h} = I_h^{2h} v^h$
$\bar{v}^2 h = v^{2h}$
$z^{2h} = I_h^{2h}(z^h - N^h v^h) + N^{2h} v^{2h}$
$v^{2h} \leftarrow NLMV^{2h}(v^{2h}, N^{2h}, z^{2h})$

(3) $v^h \leftarrow v^h + I_{2h}^h(v^{2h} - \bar{v}^{2h})$

(4) $v^h \leftarrow S2^{\nu_2}(v^h, z^h, it)$.

A complexity analysis. The main cost of our multigrid method is the cost of the smoothing steps. In our experiments we have found that it is optimal to use 3 inner Gauss-Seidel steps for each smoothing step, this makes the cost of one smoothing step $\approx 90N$, as the cost of evaluting $D_{i,j}$ for each grid point is 9 flops and the cost of one Gauss-Seidel step is 26 flops per grid point, where $N = nm$ is the total number of grid points. On Ω^{2h} there are $1/4$ the number of grid points that there are on Ω^h and in general if $p = 2^l$ there are $p^{-2} = (1/4)^l$ as many grid points on Ω^{ph} as there are on Ω^h. An upper bound on the cost of one V-cycle is therefore

$$\lim_{l\to\infty}(\nu_1+\nu_2)90N\sum_{n=0}^{l}(1/4)^n = 90(\nu_1+\nu_2)N\left(\frac{1}{1-1/4}\right) = 120(\nu_1+\nu_2)N, \tag{25}$$

which gives rise to the expected $O(N)$ complexity if a small and finite number of V-cycles is desired.

4 Refinement by Krylov acceleration

A possible way to accelerate a nonlinear method is with a Krylov acceleration scheme introduced by Oosterlee and Washio [11]. If we write our nonlinear system of equations as

$$F(u) = Nu - f = 0 \tag{26}$$

(where we have dropped the superscript h as we are referring to the fine grid only) then given a current approximation to the solution u^C resulting from the most recent multigrid step and l stored intermediate (or past) solutions $u_1, .., u_l$ (obviously if only k previous multigrid cycles have been performed where $k < l$ then we will have only k intermediate solutions) we wish to find a more optimal solution in the space

$$u^C + span[u_1 - u^C, u_2 - u^C, ..., u_l - u^C]. \tag{27}$$

In order to do this we make a linear approximation of the nonlinear operator F around u^C on the space $u^C + span[u_1 - u^C, u_2 - u^C, ..., u_l - u^C]$:

$$\begin{aligned} F(u^C + \sum_{j=1}^{l} \theta_j (u_j - u^C)) &\approx F(u^C) + \sum_{j=1}^{l} \theta_j \left(\frac{\partial F}{\partial u} \right)_{u^C} (u_j - u^C) \\ &\approx F(u^C) + \sum_{j=0}^{l-1} \theta_j (F(u_j) - F(u^C)). \end{aligned} \tag{28}$$

We then define a new solution

$$u^A = u^C + \sum_{j=1}^{l} \theta_j (u_j - u^C) \tag{29}$$

where the parameters $\theta_1, .., \theta_l$ are chosen so as to minimize

$$\| F(u^C) + \sum_{j=1}^{l} \theta_j (F(u_j) - F(u^C)) \|_2. \tag{30}$$

If we denote $F(u^C)$ by X, $\sum_{j=1}^{l} \theta_j (F(u_j) - F(u^C))$ by Y and $F(u_j) - F(u^C)$ by Z_j. Minimizing (30) is equivalent to minimizing

$$\Phi = (X + Y, X + Y) = (X, X) + 2(X, Y) + (Y, Y) \tag{31}$$

where

$$(X, Y) = \theta_1 (X, Z_1) + ... + \theta_l (X, Z_l) \tag{32}$$

and

$$(Y, Y) = \sum_{j=1}^{l} \theta_j \left[\sum_{i=1}^{l} \theta_i (Z_j, Z_i) \right]. \tag{33}$$

$$\frac{\partial \Phi}{\partial \theta_i} = 2(X, Z_i) + 2 \sum_{j=1}^{l} \theta_j (Z_i, Z_j). \tag{34}$$

Setting $\frac{\partial \Phi}{\partial \theta_i} = 0$ for all i we get an $l \times l$ linear system

$$A\theta = \sigma \tag{35}$$

to solve in order to find the optimal values for $\theta_1, .., \theta_l$, where

$$\begin{aligned} a_{ij} = (Z_i, Z_j) = (F(u_i) - F(u^C), F(u_j) - F(u^C)) = (F(u_i), F(u_j)) - (F(u^C), F(u_i)) \\ -(F(u^C), F(u_i)) - (F(u^C), F(u^C)) \end{aligned} \tag{36}$$

and

$$\sigma_i = -(X, Z_i) = -(F(u^C), F(u_i) - F(u^C)) = (F(u^C), F(u^C)) - (F(u^C), F(u_i)). \tag{37}$$

Since we are using a linear approximation of our nonlinear operator we have to take into account the fact that in some cases this approximation may not be reasonable (i.e. accurate), and that as the number of intermediate solutions used increases the accuracy of the approximation may decrease. In order to protect against this the following selection and restart criteria are proposed as in [11].

Selection Criteria. The following 2 criteria are used to decide whether u^A is a suitable solution (if not u^C is chosen):

(1) $\|F(u^A)\|_2 < \gamma_A \min(\|F(u^C)\|_2, \|F(u_1)\|_2, ..., \|F(u_l)\|_2)$

(2) $\epsilon\|u^A - u^C\|_2 < \min(\|u^A - u_1\|_2, ..., \|u^A - u_l\|_2)$
or $\|F(u^A)\|_2 < \delta \min(\|F(u^C)\|_2, \|F(u_1)\|_2, ..., \|F(u_l)\|_2)$,

where γ_A is chosen to be 2, ϵ to be 0.1 and δ to be 0.9. Condition 1 says that the residual norm of the new solution should not be considerably larger than that of the intermediate solutions and condition 2 states that u^A should not be too close to any of the intermediate solutions unless a significant reduction of the residual norm occurs.

Restart Criteria. The acceleration process is restarted (i.e all stored solutions dropped) if either of the following criteria are found in two consecutive iterations:

(1) $\|F(u^A)\|_2 \geq \gamma_B \min(\|F(u^C)\|_2, \|F(u_1)\|_2, ..., \|F(u_l)\|_2)$

(2) $\epsilon\|u^A - u^C\|_2 \geq \min(\|u^A - u_1\|_2, ..., \|u^A - u_l\|_2)$
and $\|F(u^A)\|_2 \geq \delta \min(\|F(u^C)\|_2, \|F(u_1)\|_2, ..., \|F(u_l)\|_2)$.

These conditions are just the opposite of the selection conditions. γ_B is taken as 2 (note γ_A and γ_B can take different values but γ_B must always be greater than 1).

The extra costs associated with Krylov acceleration of a multigrid step are the evaluation of several residuals and inner products, the cost of which is approximately $100N$ and the direct solution of a small linear system, the cost of which is negligible, thus the overall cost of a Krylov accelerated nonlinear multigrid step is still $O(N)$. For the case where $\nu_1 = \nu_2 = 5$ (as used in the experiments in the next section) Krylov acceleration adds approximately 10% to the total cost.

5 Numerical results

In our first experiment we compare the performance of our multigrid method when $S1$ is used as a smoother with the performance when $S2$ is used as smoother. Experiments are carried out on a gray-scale image with 256×256 pixels and range $[0, 255]$ to which gaussian white noise with SNR≈ 5 (SNR$= \sum_\Omega (u_{i,j} - \bar{u})^2 / (\sum_\Omega (n_{ij})^2)$ where $\bar{u}$ is the mean of of u) is added. The original and noisy images are shown below. In this experiment we take Ω^h to be the unit square. We take values of 30 and 0.01 for

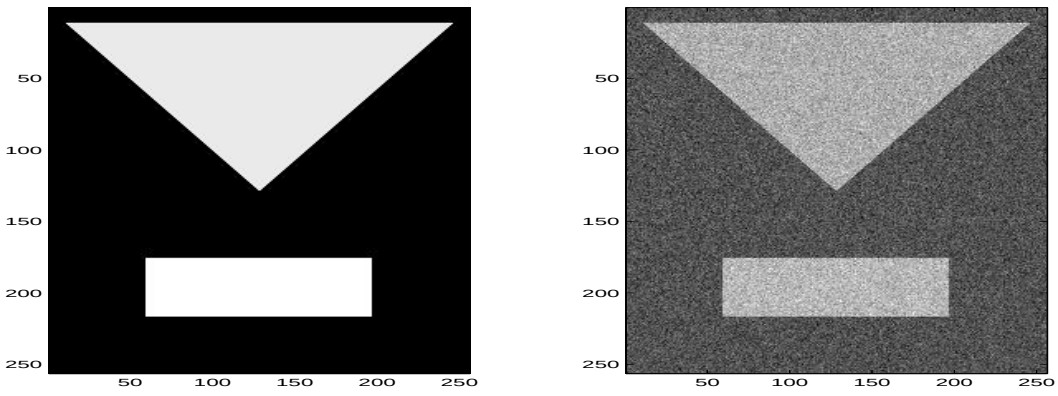


Figure 1: Original (left) and noisy (right) images

α_h and β_h. We take $\nu_1 = \nu_2 = 10$ in our multigrid method, as this appears to be optimal. The coarsest grid used is the 8×8 grid, which means we have 6 grid levels in total. The multigrid method is run with both smoothers until the relative residual has been reduced by a factor of 10^{-4}. The parameters *maxit* and *tol* in $S1$ are taken as 2 and 10^{-6} respectively. In $S2$ we perform 3 steps of Gauss-Seidel on the linear system i.e we take *it* to be 3. The table below shows the performance of the method for the two smoothers:

Smoother	Number of multigrid steps required for convergence	CPU time (s)
S1	9	8982
S2	6	221

As we can see the multigrid method with $S2$ converges in three less steps than with $S1$ and is around 40 times faster to run. The processed image from S2 is displayed in Figure 2.

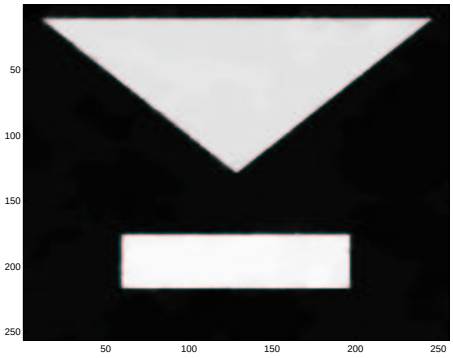


Figure 2: Image recovered using nonlinear multigrid with S2 smoother

In our second experiment we compare the performance of our nonlinear multigrid method (NMG) with that of the primal-dual Newton Method (P-D) [3] for various sizes of image. We apply both methods to various sizes of the image shown in Figure 1 (the right plot with noise). As before we have SNR≈ 5. We take values of 30 and .01 for the parameters α_h and β_h in all experiments. The nonlinear multigrid method is run with $S2$ as smoother with 3 inner steps of Gauss-Seidel, 10 pre-correction and 10 post-correction smoothing steps are used and a coarsest grid 8×8. We also run the nonlinear multigrid method with the Krylov acceleration, in this case $\nu_1 = \nu_2 = 5$ for the multigrid steps as this appears to be optimal, the number of intermediate solutions stored is 5. The inner linear steps in the primal-dual method are done using preconditioned conjugate gradient with incomplete Cholesky preconditioner, on step k the stopping criteria for these inner steps is a relative decrease of the linear residual by a factor of $\min(0.1, 0.9\|r_{k-1}\|^2/\|r_{k-2}\|^2)$ where r is the nonlinear residual, as specified in [3]. The initial guess for all methods is the noisy image z and the stopping criteria for all methods is a decrease in the relative residual by a factor of 10^{-4}.

All the results for comparison can be found in Table 1. We see that both the multigrid method on its own and the Krylov accelerated method take less cpu time to run than the Primal-Dual method, also this difference is larger as a percentage, as the size of the image increases. In the case of the 1024×1024 image the Krylov accelerated multigrid method is almost twice as fast as the primal dual method. We remark that *the optimal choice for the P-D method suggested in [3] is not strictly a Newton method, as the Newton step is not accurately done; however if a normal (or high) accuracy (say with tolerance* 10^{-4}*)*

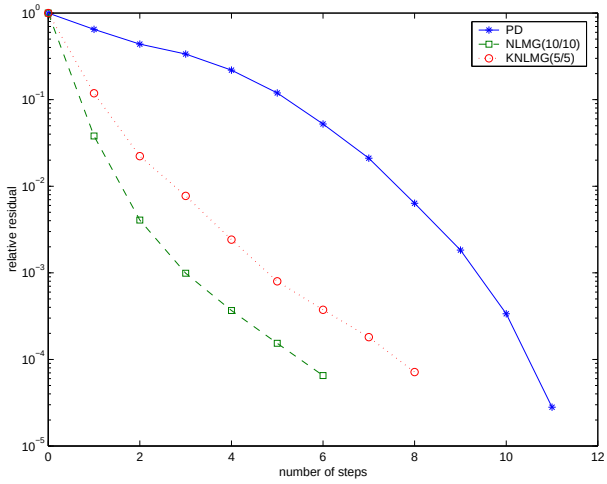
Table 1: Comparison of the proposed NMG and Krylov NMG with the P-D method

size	P-D			NLMG(10/10)			KNLMG(5/5)		
	steps	cpu time (s)	flops	steps	cpu time (s)	flops	steps	cpu time (s)	flops
256	12	235	7.5e(8)	6	221	9.5e(8)	8	148	7.0e(8)
512	14	1158	3.5e(9)	6	891	3.8e(9)	8	610	2.8(9)
1024	14	4947	1.5(10)	5	3029	1.3(10)	8	2507	1.1(10)

is used, then the P-D is not as competitive.

The convergence history of the three methods for the 256×256 case is shown in figure 3. The relative residual versus number of iterations is shown on a log scale. We see that the nonlinear multigrid method shows fastest convergence on the first step and then slows, when Krylov acceleration is used this slowing down of the convergence seems to be less dramatic. The convergence of the primal-dual method on the other hand accelerates as the number of steps increases (note that the cost of a step in the primal dual method is variable).

Figure 3: A comparison of the new Nonlinear Multigrid and its Krylov accelerated variant with the Primal Dual method.



Next we test our method on 2 more realistic images, the lenna image with SNR$\approx$ 3 and the satellite image with SNR $\approx$ 4. We then test the method on images with large amounts of noise, we use the triangle image with SNR$\approx$ 0.8 and the satellite image with SNR$\approx$ 1. All images are 256×256 in size and have range $[0, 255]$. β_h is taken as 0.01 in all cases, α_h is taken as 75 for the triangle, 30 for lenna and for the satellite with SNR 4 and 50 for the satellite with SNR 1. As before the initial guess is the noisy image and the stopping criteria is a decrease in the relative residual by a factor of 10^{-4}. The results are given below.

Image	P-D		NLMG(10/10)		KNLMG(5/5)	
	steps	cpu time (s)	steps	cpu time (s)	steps	cpu time (s)
Lenna (SNR 3)	14	288	8	288	11	209
Satellite(SNR 4)	13	263	8	304	11	205
Satellite(SNR 1)	15	361	13	497	19	349
Triangle(SNR 0.8)	13	331	9	329	13	245

Finally we comment on the performance of our multigrid method with respect to the parameter β_h (comments relate to experiments run using the 256×256 triangle image with SNR 5 and α_h fixed at 30) .

Figure 4: The original and true images (left: satellite as in [10] and right: Lenna as widely used).

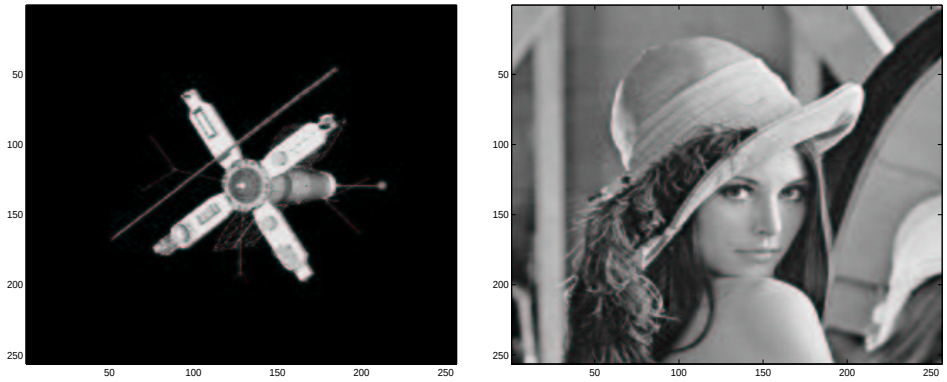


Figure 5: Noisy Satellite (SNR=4) and recovered image

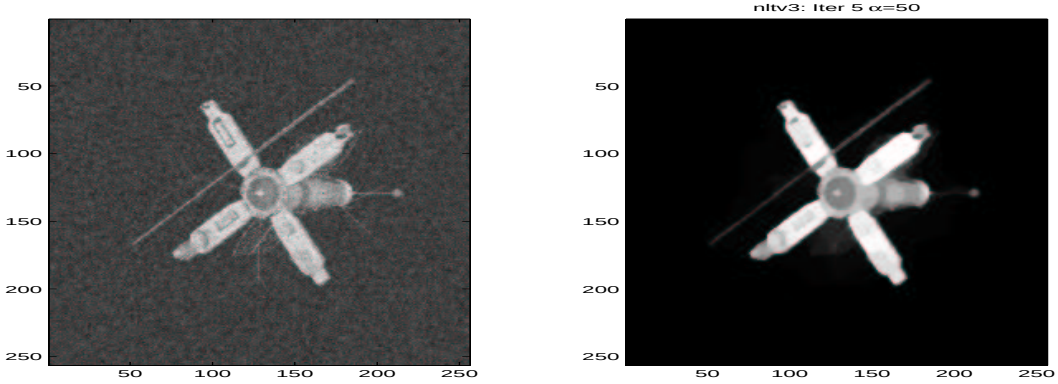


Figure 6: Noisy Lenna (SNR=3) and recovered image

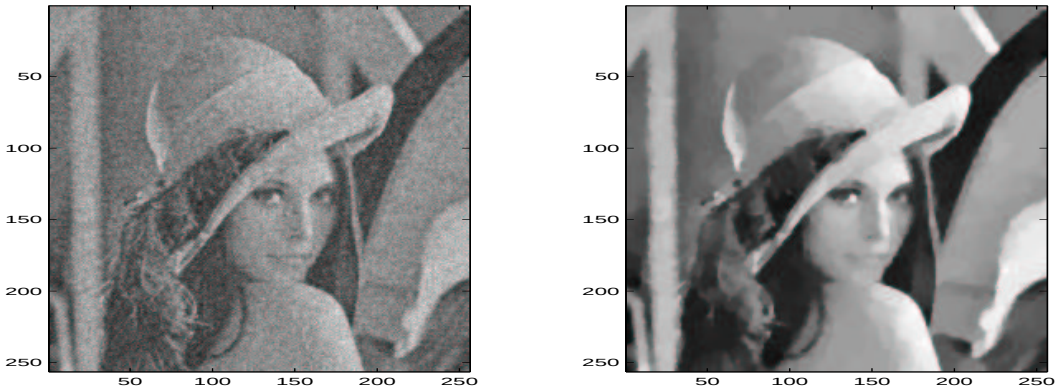


Figure 7: Noisy Triangle (SNR=0.8) and recovered image

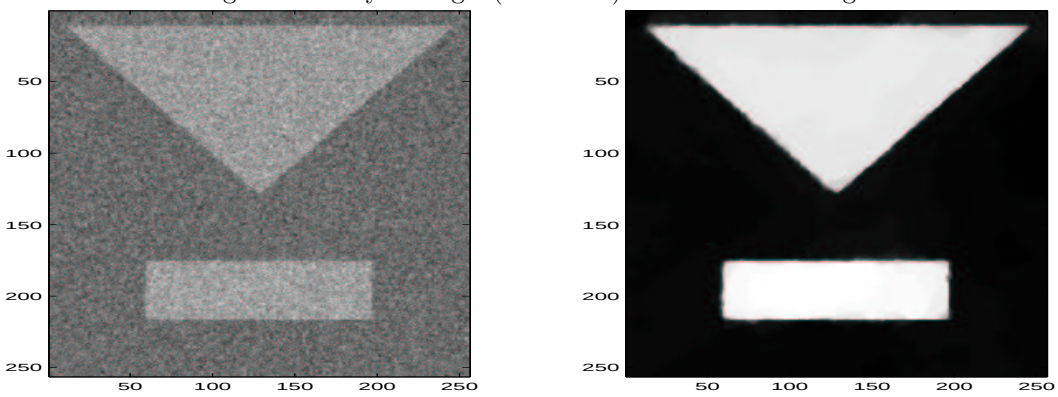
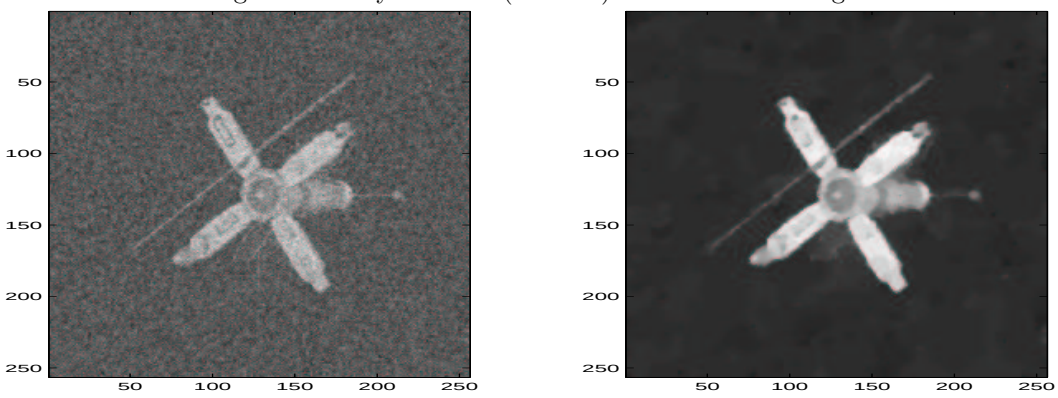


Figure 8: Noisy Satellite (SNR=1) and recovered image



The larger the value of β_h, the better our multigrid method performs. For $\beta_h = 10^{-3}$ we found that both the multigrid method and the Krylov accelerated multigrid method took roughly twice as long to run as when β_h was taken as 10^{-2} and in the case of the multigrid method on its own we had to increase the number of pre and post correction steps to achieve convergence. For values of $\beta_h \leq 10^{-4}$ the multigrid method requires further acceleration.

6 Conclusions

The standard nonlinear multigrid method does not work using the commonly-used pointwise Gauss-Seidel-Newton smoothers. Provided β is not too small, the nonlinear multigrid method with our recommended smoothers (especially S2) is competitive with the primal-dual method for a range of different images and noise levels and may offer a particular advantage for processing very larger images. The use of a Krylov acceleration procedure makes our nonlinear multigrid method faster in all our experiments. Further work is needed on our multigrid method if small values of β are desired. In addition, other ideas such as algebraic multigrids and multigrids based on optimisation may also be considered.

References

[1] W. Briggs, *A Multigrid Tutorial*, SIAM, Philadelphia, 1987.

[2] R. H. Chan, T. F. Chan, W. L. Wan, *Multigrid for Differential-Convolution Problems Arising from Image Processing*, Proceedings of the Workshop on Scientific Computing 97, Springer-Verlag, 1997.

[3] 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.

[4] T. F. Chan, H. M. Zhou, R. H. Chan, *Continuation Method for Total Variation Denoising Problems*, UCLA CAM Report, 1995.

[5] V. E. Henson, *Multigrid Methods for Nonlinear Problems: An Overview*, Center for Applied Scientific Computing Lawrence Livermore Laboratory.

[6] A. Marquina and S. Osher, *Explicit Algorithms for a New Time Dependant Model Based on Level Set Motion for Nonlinear Deblurring and Noise Removal*, SIAM J. Sci. Comput., **22** (2000), pp. 387-405.

[7] L. I. Rudin, S. Osher and E. Fatemi, *Nonlinear Total Variation Based Noise Removal Algorithms*, Physica D, **60** (1992), pp. 259-268.

[8] U. Trottenberg, C. Oosterlee, A. Schuller, *Multigrid*, Academic Press, London, 2001.

[9] C. R. Vogel and M. E. Oman, *Iterative Methods for Total Variation Denoising*, SIAM J. Sci. Comput., **17** (1996), pp. 227-238.

[10] C. R. Vogel, *Computational Methods for Inverse Problems*, SIAM, Philadelphia, 2002.

[11] T. Washio, C. Oosterlee, *Krylov Subspace Acceleration for Nonlinear Multigrid Schemes*, Electronic Transactions on Numerical Analysis., **6** (1997), pp. 271-290.

[12] P. Wesseling, *An Introduction to Multigrid Methods*, Wiley, Chichester, 1992.