



Research article

Two new preconditioners for mean curvature-based image deblurring problem

Shahbaz Ahmad¹, Adel M. Al-Mahdi^{2,*} and Rashad Ahmed³

¹ Department of Mathematics & Statistic, King Fahd University of Petroleum & Minerals, Dhahran, 31261, Saudi Arabia

² The Preparatory Year Program and The Interdisciplinary Research Center in Construction and Building Materials, King Fahd University of Petroleum & Minerals, Dhahran 31261, Saudi Arabia

³ Research Institute, King Fahd University of Petroleum & Minerals, Dhahran, 31261, Saudi Arabia

* **Correspondence:** Email: almahdi@kfupm.edu.sa.

Abstract: The mean curvature-based image deblurring model is widely used to enhance the quality of the deblurred images. However, the discretization of the associated Euler-Lagrange equations produce a nonlinear ill-conditioned system which affect the convergence of the numerical algorithms like Krylov subspace methods. To overcome this difficulty, in this paper, we present two new symmetric positive definite (SPD) preconditioners. An efficient algorithm is presented for the mean curvature-based image deblurring problem which combines a fixed point iteration (FPI) with new preconditioned matrices to handle the nonlinearity and ill-conditioned nature of the large system. The eigenvalues analysis is also presented in the paper. Fast convergence has shown in the numerical results by using the proposed new preconditioners.

Keywords: image deblurring; mean curvature; ill-posed problem; numerical analysis; preconditioning

Mathematics Subject Classification: 65N06, 65N12, 68U10, 94A08

1. Introduction

In the last two decades, the nonlinear variational methods have received a great deal of attention in the field of image deblurring. Two main difficulties arise while applying nonlinear variational techniques to large-scale noisy and blurred images. One of them, of course, is the nonlinearity and other is the solution of the large system which arise from the discretization of these such problems. The main focus of this paper is to handle these two computational difficulties. The most well-known nonlinear variational image deblurring model is total variation (TV) model [1, 10, 14]. It is one of the

most commonly used regularization model and it has nice properties like edge preserving. But the main drawback of the TV model is that the resulting images look blocky. Because this model converts smooth functions into piecewise constant functions which create staircase effects in resulting images. To reduce the staircase effects one remedy is to use the mean curvature (MC) [4, 12, 16, 18, 19] based regularization models.

The MC-based regularization models are widely used in all image processing problems. In image deblurring, the MC-based models are very effective. These models not only preserve edges but also remove staircase effect in the recovery of digital images. However, the discretization of the associated Euler-Lagrange equations produce a large nonlinear ill-conditioned system which affect the convergence of the numerical algorithms like Krylov subspace methods (GMRES etc.). Furthermore the Jacobian matrix of the MC-based nonlinear system is block banded matrix with large bandwidth. The MC-based regularization methods are effective, but due to the high nonlinearity and ill-conditioned system, robust and fast numerical solution is a crucial issue. To overcome these difficulties, in this paper we introduce two new symmetric positive definite (SPD) preconditioners. So instead of applying ordinary Generalized Minimal Residual (GMRES) method we use PGMRES (preconditioned Generalized Minimal Residual) method to solve the system. Fast convergence has shown in the numerical results by using proposed new preconditioners.

The contributions of the manuscript include the following:

- (i) our work presents an efficient algorithm for mean curvature-based image deblurring problem which combines a fixed point iteration with new SPD preconditioned matrices to handle the nonlinearity and ill-conditioned nature of the large system;
- (ii) presents a better treatment for the computationally expensive higher order and nonlinear mean curvature regularization functional.

The paper is organized in different sections. The first section includes introduction while the second section includes problem description of image deblurring model. In the third section, we present nonlinear system of first order equations, cell discretization and CCFD method. In fourth section, we introduce our proposed SPD preconditioners and our algorithmic technique. The properties and eigenvalue analysis of the proposed preconditioned matrices are discussed in the fourth section. The numerical experiments are also presented in the fourth section. The conclusions about the proposed PGMRES method is discussed in the last section of the paper.

2. Problem description

The focus of the paper is on image deblurring problem, so we start by presenting its concise description. Mathematically, the relationship between u (original image) and z (recorded image) is as follows;

$$z = \vec{K}u + \epsilon, \quad (2.1)$$

where ϵ is the noise function. The noise can be Gaussian noise, salt and pepper noise, Brownian noise etc. In this paper, we have considered Gaussian noise. The \vec{K} is the blurring operator which is a Fredholm-integral operator of first kind;

$$(\vec{K}u)(x) = \int_{\Omega} k(x, y)u(y) dy, \mathbf{x} \in \Omega,$$

having translation invariance property i.e., $k(x, y) = k(x - y)$. If $\vec{K} = I$ problem (2.1) is called image denoising problem. The problem (2.1) becomes ill-posed [1, 13, 14] because \vec{K} is a compact operator. Let Ω be a square in \mathbb{R}^2 and u be an image intensity function defined on Ω . The $\mathbf{x} = (x, y)$ defines the position in Ω . Let $|\mathbf{x}| = \sqrt{x^2 + y^2}$ is an Euclidean norm and $\|\cdot\|$ is $L_2(\Omega)$ norm. The problem (2.1) is an inverse problem. The recovering of u from z makes (2.1) an unstable problem [1, 13, 14]. To make it stable, one remedy is to use the MC regularization functional [4, 12, 16, 18, 19],

$$J(u) = \int_{\Omega} \kappa(u)^2 \mathbf{d}\mathbf{x} = \int_{\Omega} (\nabla \cdot \frac{\nabla u}{|\nabla u|})^2 \mathbf{d}\mathbf{x}.$$

where κ is called the mean curvature surface. Then the problem (2.1) takes the form, find u that minimizes the following problem

$$T(u) = \frac{1}{2} \|\vec{K}u - z\|^2 + \frac{\alpha}{2} J(u), \quad (2.2)$$

where $\alpha > 0$ is a regularization parameter. The well-posedness of the problem (2.2) for a particular case (synthetic image denoising problem) is explained in [18]. Then the Euler-Lagrange equations associated with (2.2) are,

$$\vec{K}^*(\vec{K}u - z) + \alpha \nabla \cdot \left[\frac{\nabla \kappa}{\sqrt{|\nabla u|^2 + \beta^2}} - \frac{\nabla \kappa \cdot \nabla u}{(\sqrt{|\nabla u|^2 + \beta^2})^3} \nabla u \right] = 0 \text{ in } \Omega, \quad (2.3)$$

$$\frac{\partial u}{\partial n} = 0 \text{ in } \partial\Omega, \quad (2.4)$$

$$\kappa(u) = 0 \text{ in } \partial\Omega, \quad (2.5)$$

where \vec{K}^* denotes the adjoint operator of \vec{K} and $\beta > 0$ is used to avoid non-differentiability at zero. The equation (2.3) is a nonlinear fourth order partial differential equation.

As it is known that the MC-based model does not only preserve edges but also removes staircase effect in the recovery of digital images. However, fourth order derivatives appear in the Euler-Lagrange equations, which create problems in developing an efficient numerical algorithm. One key problem in presenting the method is to give a proper approximation to the nonlinear mean curvature functional. We have treated this difficulty by reducing the nonlinear fourth order Euler-Lagrange equation into a system of first order equations.

3. The first order nonlinear system

The Eq (2.3) can be expressed as first order nonlinear system,

$$\vec{K}^* \vec{K}u + \alpha \nabla \cdot \vec{p} - \alpha \nabla \cdot \vec{t} = \vec{K}^* z, \quad (3.1)$$

$$-w + \nabla \cdot \vec{v} = 0, \quad (3.2)$$

$$\sqrt{|\nabla u|^2 + \beta^2} \vec{v} - \nabla u = 0, \quad (3.3)$$

$$\sqrt{|\nabla u|^2 + \beta^2} \vec{p} - \nabla w = 0, \quad (3.4)$$

$$\sqrt{|\nabla u|^2 + \beta^2} \vec{t} - (\nabla w \cdot \vec{v}) \vec{v} = 0, \quad (3.5)$$

where

$$\vec{v} = \frac{\nabla u}{\sqrt{|\nabla u|^2 + \beta^2}}, \quad w = \nabla \cdot \vec{v}, \quad \vec{p} = \frac{\nabla w}{\sqrt{|\nabla u|^2 + \beta^2}} \quad \text{and} \quad \vec{t} = \frac{(\nabla w \cdot \vec{v}) \vec{v}}{\sqrt{|\nabla u|^2 + \beta^2}}.$$

We will take advantage of this special structure to derive our proposed algorithm.

3.1. Cell discretization

For MC-based image deblurring problem, the domain $\Omega = (0, 1) \times (0, 1)$ is partitioned by $\delta_x \times \delta_y$, [11], where

$$\begin{aligned} \delta_x : 0 &= x_{1/2} < x_{3/2} < x_{5/2} < \dots < x_{n_x-1/2} < x_{n_x+1/2} = 1, \\ \delta_y : 0 &= y_{1/2} < y_{3/2} < y_{5/2} < \dots < y_{n_x-1/2} < y_{n_x+1/2} = 1, \end{aligned}$$

where n_x represents the number of equispaced partitions in the x or y directions and (x_i, y_j) denotes centers of the cells. The

$$\begin{aligned} x_i &= (i - \frac{1}{2})h \quad i = 1, 2, 3, \dots, n_x, \\ y_j &= (j - \frac{1}{2})h \quad j = 1, 2, 3, \dots, n_x, \end{aligned}$$

where $h = \frac{1}{n_x}$. The $(x_{i \pm \frac{1}{2}}, y_j)$ and $(x_i, y_{j \pm \frac{1}{2}})$ are representing midpoints of cell edges,

$$\begin{aligned} x_{i \pm \frac{1}{2}} &= x_i \pm \frac{h}{2} \quad i = 1, 2, 3, \dots, n_x, \\ y_{j \pm \frac{1}{2}} &= y_j \pm \frac{h}{2} \quad j = 1, 2, 3, \dots, n_x. \end{aligned}$$

The set

$$e_{ij} = \left\{ (x, y) : x \in [x_i - \frac{1}{2}, x_i + \frac{1}{2}], y \in [y_j - \frac{1}{2}, y_j + \frac{1}{2}] \right\},$$

represents a cell with (x_i, y_j) as a center. Let

$$\begin{aligned} \chi_i(x) &= \begin{cases} 1 & x \in (x_i - \frac{1}{2}, x_i + \frac{1}{2}) \\ 0 & \text{otherwise,} \end{cases} \\ \chi_j(y) &= \begin{cases} 1 & y \in (y_i - \frac{1}{2}, y_i + \frac{1}{2}) \\ 0 & \text{otherwise.} \end{cases} \end{aligned}$$

And

$$\begin{aligned} \phi_i(x_l + \frac{1}{2}) &= \delta_{il}, \\ \phi_k(y_j + \frac{1}{2}) &= \delta_{jk}. \end{aligned}$$

Approximations of u and w are

$$u(x, y) \cong \bar{U}(x, y) = \sum_{i=1}^{n_x} \sum_{j=1}^{n_x} u_{ij} \chi_i(x) \chi_j(y),$$

and

$$w(x, y) \cong \bar{W}(x, y) = \sum_{i=1}^{n_x} \sum_{j=1}^{n_x} w_{ij} \chi_i(x) \chi_j(y),$$

respectively, where $u_{ij} = \bar{U}(x_i, y_j)$ and $w_{ij} = \bar{W}(x_i, y_j)$. The representation of the data z is

$$z(x, y) \cong \bar{Z}(x, y) = \sum_{i=1}^{n_x} \sum_{j=1}^{n_x} z_{ij} \chi_i(x) \chi_j(y),$$

where z_{ij} can be calculated at cell averages. By applying midpoint quadrature approximation, we have

$$(Ku)(x_i, y_j) \cong [K_h U]_{(ij)}.$$

Denote $\vec{v} = (v^x, v^y)$, $\vec{p} = (p^x, p^y)$ and $\vec{t} = (t^x, t^y)$. The approximation of x and y components of \vec{v} are

$$\bar{V}^x(x, y) = \sum_{i=1}^{n_x-1} \sum_{j=1}^{n_x} v_{ij}^x \phi_i(x) \chi_j(y), \quad \bar{V}^y(x, y) = \sum_{i=1}^{n_x-1} \sum_{j=1}^{n_x} v_{ij}^y \phi_i(y) \chi_j(x),$$

respectively. $\bar{V} = [\bar{V}^x \ \bar{V}^y]^t$ denotes the discretization of \vec{v} . Similarly, approximation of the components of \vec{p} and \vec{t} are

$$\bar{P}^x(x, y) = \sum_{i=1}^{n_x-1} \sum_{j=1}^{n_x} p_{ij}^x \phi_i(x) \chi_j(y), \quad \bar{P}^y(x, y) = \sum_{i=1}^{n_x-1} \sum_{j=1}^{n_x} p_{ij}^y \phi_i(y) \chi_j(x),$$

and

$$\bar{T}^x(x, y) = \sum_{i=1}^{n_x-1} \sum_{j=1}^{n_x} t_{ij}^x \phi_i(x) \chi_j(y), \quad \bar{T}^y(x, y) = \sum_{i=1}^{n_x-1} \sum_{j=1}^{n_x} t_{ij}^y \phi_i(y) \chi_j(x),$$

respectively. The $\bar{P} = [\bar{P}^x \ \bar{P}^y]^t$ and $\bar{T} = [\bar{T}^x \ \bar{T}^y]^t$ denote the discretization of the vectors \vec{p} and \vec{t} respectively.

3.2. The CCFD method

Here, we present the cell-centered finite difference (CCFD) method for MC-based image deblurring problem. By considering lexicographical ordering of the unknowns,

$$\begin{aligned} U &= [\bar{U}_{11} \quad \dots \quad \bar{U}_{n_x n_x}]^t, & W &= [\bar{W}_{11} \quad \dots \quad \bar{W}_{n_x n_x}]^t, \\ V &= [\bar{V}_{11}^x \quad \dots \quad \bar{V}_{n_x-1 n_x-1}^x \quad \bar{V}_{11}^y \quad \dots \quad \bar{V}_{n_x-1 n_x-1}^y]^t, \\ P &= [\bar{P}_{11}^x \quad \dots \quad \bar{P}_{n_x-1 n_x-1}^x \quad \bar{P}_{11}^y \quad \dots \quad \bar{P}_{n_x-1 n_x-1}^y]^t, \end{aligned}$$

$$\text{and } T = [\bar{T}_{11}^x \quad \dots \quad \bar{T}_{n_x-1, n_x-1}^x \quad \bar{T}_{11}^y \quad \dots \quad \bar{T}_{n_x-1, n_x-1}^y]^t.$$

Now by applying CCFD method to (3.1)–(3.5) we obtain the following system,

$$K^* K_h U - \alpha A_h W + \alpha B_h^* P - \alpha B_h^* T = K_h^* Z, \quad (3.6)$$

$$-I_h W + B_h^* V = O, \quad (3.7)$$

$$D_h V + B_h U = O, \quad (3.8)$$

$$D_h P + B_h W = O, \quad (3.9)$$

$$D_h T - C_h V = O, \quad (3.10)$$

where midpoint quadrature rule is used for the integral term. The matrices K_h, A_h and I_h (the identity matrix) each one is of size $n_x^2 \times n_x^2$. The matrix B_h is of size $2n_x(n_x - 1) \times n_x^2$. The matrices C_h and D_h are of size $2n_x(n_x - 1) \times 2n_x(n_x - 1)$. So we have the following system

$$\begin{bmatrix} K_h^* K_h & -\alpha A_h & O & \alpha B_h^* & -\alpha B_h^* \\ O & -I_h & B_h^* & O & O \\ B_h & O & D_h & O & O \\ O & B_h & O & D_h & O \\ O & O & -C_h & O & D_h \end{bmatrix} \begin{bmatrix} U \\ W \\ V \\ P \\ T \end{bmatrix} = \begin{bmatrix} K_h^* Z \\ O \\ O \\ O \\ O \end{bmatrix}.$$

The matrix K_h is block Toeplitz with Toeplitz blocks (BTTB) and $K_h^* K_h$ is SPD. The matrix A_h is a diagonal matrix having following structure,

$$A_h = \frac{2}{\beta h} (A_1 + A_2),$$

where both A_1 and A_2 are of size $n_x^2 \times n_x^2$.

$$A_1 = \tilde{I} \otimes E \quad \text{and} \quad A_2 = E \otimes \tilde{I},$$

where \otimes is a tensor product. The size of identity matrix \tilde{I} is $n_x \times n_x$. The matrix

$$E = \begin{bmatrix} 1 & & & & \\ & 0 & & & \\ & & \ddots & & \\ & & & 0 & \\ & & & & 1 \end{bmatrix},$$

is of size $n_x \times n_x$. The matrix B_h has the following structure,

$$B_h = \frac{1}{h} \begin{bmatrix} B_1 \\ B_2 \end{bmatrix},$$

where both B_1 and B_2 are of size $n_x(n_x - 1) \times n_x^2$, and

$$B_1 = F \otimes \tilde{I} \quad \text{and} \quad B_2 = \tilde{I} \otimes F, \text{ where}$$

$$F = \begin{bmatrix} 1 & -1 & & & & \\ & 1 & -1 & & & \\ & & \ddots & \ddots & & \\ & & & \ddots & -1 & \\ & & & & 1 & -1 \end{bmatrix},$$

is a matrix of size $(n_x - 1) \times n_x$. The matrix

$$C_h = \begin{bmatrix} C^x & 0 \\ 0 & C^y \end{bmatrix},$$

is a diagonal matrix and its entries are obtained by the discretization of the expression $(\nabla_w \cdot \vec{v})$. The matrix C^x is of size $(n_x - 1) \times n_x$, and the matrix C^y is of size $n_x \times (n_x - 1)$. The matrix D_h is also a diagonal matrix with positive diagonal entries and its entries are obtained by the discretization of the expression $\sqrt{|\nabla u|^2 + \beta^2}$. The matrix D_h has the following structure,

$$D_h = \begin{bmatrix} D^x & 0 \\ 0 & D^y \end{bmatrix},$$

where D^x is of size $(n_x - 1) \times n_x$, and D^y is of size $n_x \times (n_x - 1)$. Note that on horizontal and vertical edges of each cell e_{ij} , the values of the all unknowns are not available, so average operators are used to calculate their values.

Now if we eliminate W, V, P and T from (3.6)–(3.10), then we have the following nonlinear primal system,

$$(K_h^* K_h + \alpha L_h(U))U = K_h^* Z, \quad (3.11)$$

where

$$L_h = (B_h^* D_h^{-1} B_h)^2 + A_h (B_h^* D_h^{-1} B_h) + B_h^* D_h^{-1} C_h D_h^{-1} B_h.$$

The L_h is block pentadiagonal matrix according to the lexicographical ordering of the unknowns. The main diagonal blocks are pentadiagonal while the off-diagonal blocks are tridiagonal matrices.

In the literature, one can find a number of numerical methods that have been investigated to mean curvature-based nonlinear image minimization problems [2, 4, 8, 12, 14–16, 18, 19]. Since MC-based models produce a large and ill-conditioned nonlinear system, so almost all these standard methods get quite slow convergence. Furthermore the presence of higher order and nonlinear mean curvature regularization functional in the governing equation of the models makes these highly nonlinear systems harder for calculation. MC is very much computationally expensive, that is why, most of the existing methods performs quite poorly. To make a robust numerical method for MC-based nonlinear image deblurring problem, now we present a new preconditioned numerical method.

4. Numerical implementation

Here we introduce the algorithms to solve the MC-based nonlinear system (3.11). First we apply discrete version of the FPI (fixed point iteration) to (3.11) to handle the nonlinearity of MC. The

approach taken here is called “lagged diffusivity” scheme [14]. Its rate is just linear but in practice it has a quite rapid convergence. Furthermore, this scheme does not depend on initial guess to converge globally. This is why globalization is not an issue for this scheme. So by FPI we have a following linear system;

$$(K_h^* K_h + \alpha L_h(U^m))U^{m+1} = K_h^* Z. \quad (4.1)$$

4.1. Properties

Before proceeding further, we discuss the some important properties of our system (4.1).

- (1) The Hessian matrix $K_h^* K_h + \alpha L_h$ is extremely large for practical application. When α is small, the Hessian matrix becomes quite ill-conditioned. This happens because the eigenvalues of the blurring operator \vec{K} cluster to zero [14].
- (2) The first term, $K_h^* K_h$, in the Hessian matrix is symmetric positive definite. Although, $K_h^* K_h$ is full but the blurring operator \vec{K} has translation invariant property, which permit the use of FFT (Fast Fourier transformation) to evaluate $K_h^* K_h u$ in $O(n \log n)$ operations [14].
- (3) The second term, L_h , in the Hessian matrix is sparse but not symmetric. The L_h in (3.11) consists of three terms. The first and the last term in L_h are symmetric positive semidefinite [14] but the middle term is not symmetric. Hence the system (4.1) is not symmetric positive definite.

4.2. The Preconditioner

According to the properties of our system (4.1), mentioned above, GMRES (Generalized Minimal Residual) method, is appropriate for the solution of the system (4.1). Due to ill-conditioned system, GMRES can be quite slow to convergence. So we use preconditioned Generalized Minimal Residual (PGMRES) method. For effective solution, preconditioning matrix P , must be symmetric positive definite (SPD). Here we introduce following two SPD preconditioned matrices namely P_1 and P_2 . For the first preconditioner P_1 , we consider the following matrix

$$L_h^{TV} = B_h^* D_h^{-1} B_h. \quad (4.2)$$

The L_h^{TV} is the SPD matrix lies in the first term of L_h matrix (4.1). In fact, L_h^{TV} arise by the discretization of total variation (TV) based image deblurring problem [14]. So we introduce preconditioned matrix P_1 as

$$P_1 = \alpha I_h + \gamma \text{diag}(L_h^{TV}), \quad (4.3)$$

where I_h is an identity matrix and $\text{diag}(L_h^{TV})$ is a diagonal matrix and its entries are the diagonal entries of matrix L_h^{TV} . The γ is the positive parameter. For the second preconditioner P_2 , we consider the following SPD matrix

$$L_h^s = (B_h^* D_h^{-1} B_h)^2 + B_h^* D_h^{-1} C_h D_h^{-1} B_h. \quad (4.4)$$

The L_h^s is the SPD part (first and the third term) of L_h matrix (4.1). So the second preconditioned matrix is

$$P_2 = \alpha I_h + \gamma \text{diag}(L_h^s), \quad (4.5)$$

where $\text{diag}(L_h^s)$ is also a diagonal matrix whose entries are the diagonal entries of matrix L_h^s . While applying PGMRES to (4.1), the inversion of preconditioner matrix, will be required. Since the both terms in our preconditioning matrices (P_1 and P_2) are diagonal matrices, so inversion can be done easily. We have summarize the PGMRES method in the following algorithm.

Algorithm 1: The PGMRES Method

On mesh Ω_h ,

Step-I: Initial iteration U^0 ,

Step-II: for $m = 1; \max$

$$A^m = K_h^* K_h + \alpha L_h(U^m), b^m = K_h^* Z,$$

Step-III: Use PGMRES to solve

$$A^m U^{m+1} = b^m \quad \text{with preconditioned matrix } P,$$

where $P = P_1, P_2$.

end

4.3. Eigenvalues

Now, let the eigenvalues of $K_h^* K_h$ and L_h be λ_i^K and λ_i^L respectively such that $\lambda_i^K \downarrow 0$ and $\lambda_i^L \uparrow \infty$. So the eigenvalues of $P_1^{-1} \bar{A}$ and $P_2^{-1} \bar{A}$ are

$$\theta_i^1 = \frac{\lambda_i^K + \alpha \lambda_i^L}{\alpha \lambda_i^K + \gamma \lambda_i^{L^{TV}}} \quad \text{and} \quad \theta_i^2 = \frac{\lambda_i^K + \alpha \lambda_i^L}{\alpha \lambda_i^K + \gamma \lambda_i^{L^{ns}}}, \quad (4.6)$$

respectively. Here $\lambda_i^{L^{TV}}$ are eigenvalues of L_h^{TV} and $\lambda_i^{L^{ns}}$ are eigenvalues of L_h^{ns} . The $\bar{A} = K_h^* K_h + \alpha L_h$ is the Hessian matrix of the system (4.1). One can notice that $\lambda_i^{L^{TV}} \leq \lambda_i^{L^{ns}} \leq \lambda_i^L$. So for $\gamma > \alpha$,

$$\theta_i^1, \theta_i^2 \rightarrow 1 \quad \text{as} \quad i \rightarrow \infty. \quad (4.7)$$

Hence, for large values of γ , $P^{-1} \bar{A}$ has more favourable spectrum as compared to the Hessian matrix \bar{A} , so PGMRES converge more rapidly than ordinary GMRES. This can also be shown in the numerical examples that PGMRES is getting rapid convergence with large $\gamma > \alpha$.

4.4. Numerical experiments

Now we present four numerical examples for MC-based image deblurring problem. We have used different values of n_x so the resulting system has n_x^2 unknowns. For numerical computations, MATLAB software is used to obtain the numerical results on Intel(R) Core(TM) i7-4510U with CPU @ 2.00 GHz 2.60 GHz. To measure the quality of the deblurred images, we have used PSNR (Peak Signal to Noise Ratio) and SSIM (Structured Similarity Index Measure).

Initial guess. Since the method is globally convergent, the fixed point iteration (FPI) can be started with any initial guess [14]. In all experiments we have used $u^0 = z$ (the blurred data) as initial guess. For instance, taking the zero initial guess $u^0 = 0$ means that one extra FPI may be required to attain the accuracy obtained with $u^0 = z$.

Selection of the parameters. Although the automated selection of the parameter α is beyond the scope of the present paper, that information can be found in [3, 18, 19]. The purpose of α is to select the amount of blur / noise to be removed, so there is no point in choosing either a very large or very small α . The optimum range of α for MC-based model lies between $1e - 3$ and $1e - 8$. To understand the performance of the parameter β on our algorithm (GMRES), we have run the algorithm with fixed $\alpha = 1e - 8$ on the benchmark Cameraman image of size 256×256 . In this experiment we have varied β from $1e - 3$ to 1 and found that the smaller value of β does not improve the PSNR and SSIM. The results of this experiments summarized in Table 1.

Table 1. Results of fixed α and varying β .

| β | $1e - 4$ | $1e - 3$ | $1e - 2$ | $1e - 1$ | 1 |
|----------------|----------|----------|----------|----------|---------|
| Blurred PSNR | 22.4789 | 22.4789 | 22.4789 | 22.4789 | 22.4789 |
| Deblurred PSNR | 28.7719 | 28.7726 | 29.8719 | 30.4515 | 30.4953 |
| Deblurred SSIM | 0.7112 | 0.7119 | 0.7242 | 0.7306 | 0.7312 |

Example 1. In this example we have used Cameraman image. This is a complicated image, because it has a small scale texture part (shirt) and large scale cartoon part (face). The different aspects of Cameraman image are presented in Figure 1. The size of each subfigure is 256×256 . These are (a) blurry image (b) deblurred image by GMRES (c) deblurred image by P_1 GMRES with $\gamma = 1000$ and (d) deblurred image by P_2 GMRES with $\gamma = 1000$. The calculation of relative residual at each iterations against different values of the parameter γ is presented in Figure 2. The $f_{special}('gaussian', [n_x \ n_x], 1.5)$ kernel is used for numerical calculations. It is a *gaussian* filter of size $n_x \times n_x$ with standard deviation ($\sigma = 1.5$). The parameters $\alpha = 1e - 8$ and $\beta = 1$. For the stopping criteria of a numerical method we have used tolerance $tol = 1e - 2$.

Remarks

- (1) The Figures 1(b)–(d) are almost similar, this means both GMRES method (without preconditioning) and PGMRES (P_1 GMRES and P_2 GMRES) method are generating same quality results.
- (2) From the Figure 2, one can clearly observe the effectiveness of preconditioning. Here, result were presented using fixed point iteration count $m = 1$. The number of PGMRES iteration is much lesser than as compared to GMRES (without preconditioning) to reach the required accuracy $tol = 1e - 2$. The results for subsequent fixed point iterations are also similar.
- (3) From the Figure 2, this can also be observed that PGMRES is getting rapid convergence for large values of γ for both preconditioners (P_1 and P_2).
- (4) From Table 2, one can observe that the PSNR and SSIM by PGMRES method is same as compared with the ordinary GMRES method. But PGMRES method is producing same PSNR and SSIM in quite less iterations. For example, to achieve the same PSNR and SSIM, the P_1 GMRES method needs only 80 iterations with $\gamma = 10$ and the P_2 GMRES method needs only 42 iterations with $\gamma = 10$. But the ordinary GMRES method needs more than 150 iterations to get the same PSNR and SSIM. The number of iterations further decreases with higher values of γ . Which means that PGMRES method is faster than the ordinary GMRES method.
- (5) From Table 2, it is also observed that for both preconditioners (P_1 and P_2) the PSNR and SSIM do not get improvement with the increase in the value of γ . The higher value of γ only decreases

the number of iterations for both preconditioners (P_1 and P_2). But the decrease in the number of iterations is not much significant for the value of γ between 500 and 1000. So we may suggest that suitable value of γ lies between 500 and 1000.

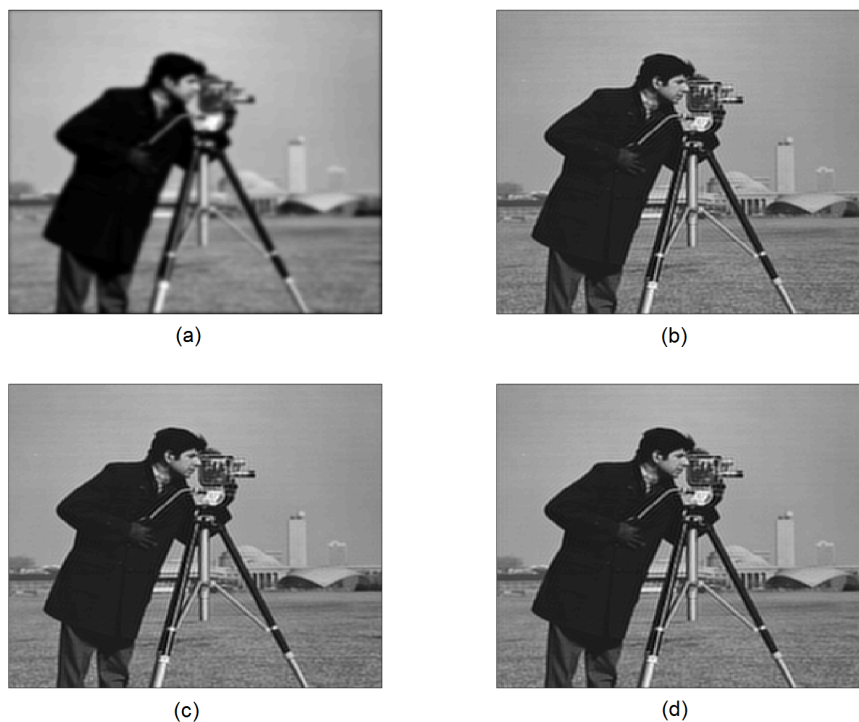


Figure 1. Cameraman image: (a) blurry image; (b) deblurred image by GMRES; (c) deblurred image by P_1 GMRES with $\gamma = 1000$; (d) deblurred image by P_2 GMRES with $\gamma = 1000$.

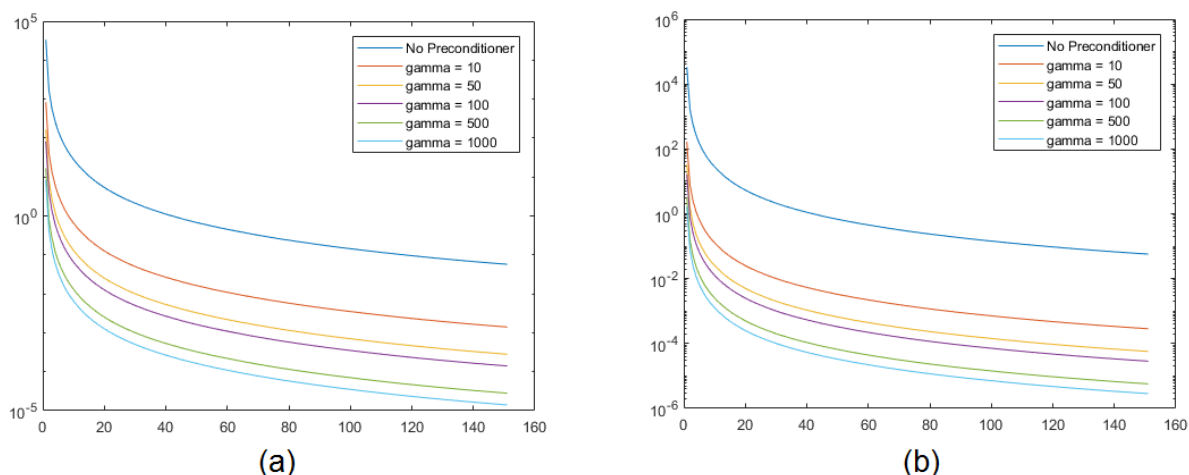


Figure 2. GMRES and PGMRES convergence for the solution to the linear system in Example 1 at fixed point iteration $m = 1$. The horizontal axis represents the number of iterations. The norm of the residual at each iteration is presenting on vertical axis. Blue line represents GMRES iterations. (a) PGMRES iteration using preconditioner P_1 against different values of γ and (b) PGMRES iteration using preconditioner P_2 against different values of γ .

Table 2. Comparison of GMRES and PGMRES for Example 1.

| Method | Blurred PSNR | γ | Deblurred PSNR | Deblurred SSIM | Iterations |
|-------------|--------------|----------|----------------|----------------|------------------|
| GMRES | 22.4789 | | 30.4953 | 0.7312 | 150 ⁺ |
| P_1 GMRES | 22.4789 | 10 | 30.5177 | 0.7362 | 80 |
| | 22.4789 | 50 | 30.5177 | 0.7362 | 40 |
| | 22.4789 | 100 | 30.5177 | 0.7362 | 24 |
| | 22.4789 | 500 | 30.5177 | 0.7362 | 16 |
| | 22.4789 | 1000 | 30.5177 | 0.7362 | 10 |
| P_2 GMRES | 22.4789 | 10 | 30.4564 | 0.7216 | 42 |
| | 22.4789 | 50 | 30.4564 | 0.7216 | 22 |
| | 22.4789 | 100 | 30.4564 | 0.7216 | 18 |
| | 22.4789 | 500 | 30.4564 | 0.7216 | 10 |
| | 22.4789 | 1000 | 30.4564 | 0.7216 | 8 |

Example 2. In this example we have used Goldhills image. This is real and synthetic image. Here we have compared our MC based algorithm with TV (total variation) based algorithm. Since TV-based model generates a SPD matrix system, so for the solution we have used CG (Conjugate Gradient) method. The different aspects of Goldhills image are presented in Figure 3. The size of each subfigure is 512×512 . These are (a) exact image (b) blurry image (c) deblurred image by CG (d) deblurred image by GMRES (e) deblurred image by P_1 GMRES and (f) deblurred image P_2 GMRES. For numerical calculations, we have used the $ke_gen(n_x, 300, 5)$ kernel [5–7, 9]. It is a circular *gaussian* filter of size

$n_x \times n_x$ with radius ($r = 300$) standard deviation ($\sigma = 5$). The $ke_gen(120, 40, 4)$ kernel is depicted in Figure 4. For TV-based method we have used $\alpha = 5e - 5$ and $\beta = 1$ according to [14]. For MC-based method we have used $\alpha = 1e - 8$ and $\beta = 1$ and $\gamma = 1000$. For the stopping criteria of a numerical methods we have used tolerance $tol = 1e - 4$.



Figure 3. Goldhills image: (a) exact image; (b) blurry image; (c) deblurred image by CG; (d) deblurred image by GMRES; (e) deblurred image by P_1 GMRES; (f) deblurred image P_2 GMRES.

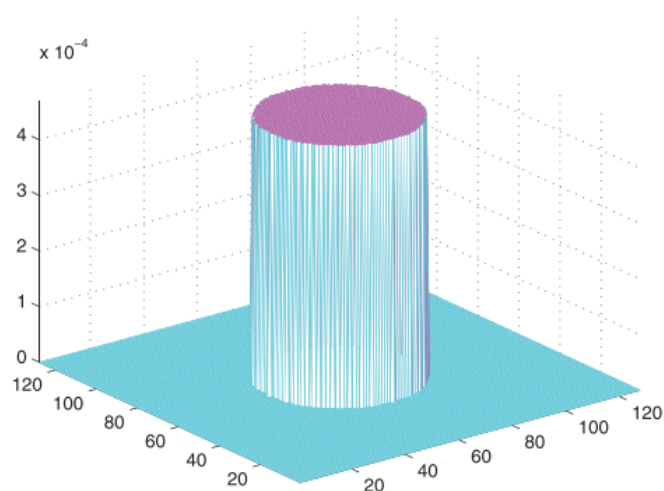


Figure 4. The blurring kernel $ke_gen(120, 40, 4)$.

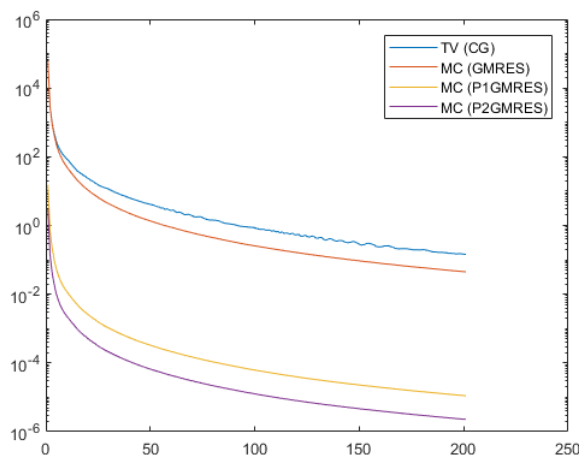


Figure 5. The CG, GMRES and PGMRES convergence for the solution to the linear system in Example 2 at fixed point iteration $m = 1$. The horizontal axis represents the number of iterations. The norm of the residual at each iteration is presenting on vertical axis. Blue line represents CG iterations, orange line represents GMRES iterations, yellow line represents P_1 GMRES iteration and purple line represents P_2 GMRES iteration.

Table 3. Comparison of CG, GMRES and PGMRES for Example 2.

| Method | CG | GMRES | P_1 GMRES | P_2 GMRES |
|----------------|---------|---------|-------------|-------------|
| Blurred PSNR | 23.7660 | 23.7660 | 23.7660 | 23.7660 |
| Deblurred PSNR | 37.4867 | 39.2180 | 39.4430 | 39.2734 |
| Deblurred SSIM | 0.9188 | 0.9272 | 0.9328 | 0.9239 |
| Iterations | 200+ | 200+ | 90 | 40 |

Remarks

- (1) From Table 3, it is observed that the PSNR and SSIM by MC-based (GMRES and PGMRES) methods are little higher than TV-based CG method. Same comparison can be observed from Figures 3(c)–(f). So MC-based (GMRES and PGMRES) methods are generating better quality results.
- (2) From the Figure 5, one can clearly observe the effectiveness of preconditioning. The number of MC-based PGMRES iteration is much lesser than as compared to both MC-based GMRES and TV-based CG methods to reach the required accuracy $tol = 1e - 4$. The results for subsequent fixed point iterations are also similar.
- (3) From Table 3, it is observed that the PSNR and SSIM by MC-based PGMRES method is almost same as compared with the ordinary MC-based GMRES method. But PGMRES method is generating same PSNR and SSIM in quite less iterations. To achieve the same PSNR and SSIM, the P_1 GMRES method needs only 90 iterations and the P_2 GMRES method needs only 40 iterations. But the GMRES method needs more than 200 iterations to get the same PSNR and SSIM. The TV-based CG method is also getting more than 200 iterations to get its PSNR and SSIM.

SSIM. Which means that MC-based PGMRES method is faster than the MC-based GMRES and TV-based CG method for real and synthetic images. Here, the performance of the preconditioner P_2 is again more effective than the preconditioner P_1 .

Example 3. Here we have used nontexture Moon image. Here we have also compared our MC based algorithm with TV (total variation) based algorithm. For TV we have used CG (Conjugate Gradient) method. The different aspects of Moon image are presented in Figure 6. The size of each subfigure is 256×256 . These are (a) exact image (b) blurry image (c) deblurred image by CG (d) deblurred image by GMRES (e) deblurred image by P_1 GMRES and (f) deblurred image P_2 GMRES. For numerical calculations, we have also used the $ke_gen(n_x, 300, 5)$ kernel. For TV-based method we have used $\alpha = 1e - 4$ and $\beta = 1$ according to [14]. For MC-based method we have used $\alpha = 1e - 6, \beta = 1$ and $\gamma = 1000$. For comparison we have used three different values of n_x . These are 64, 128 and 256. For the stopping criteria of a numerical methods we have used tolerance $tol = 1e - 5$.

Remarks

- (1) The Figures 6(b)–(d) are almost similar, so all methods are generating same quality results.
- (2) From the Figure 7, one can clearly notice the effectiveness of preconditioning. For all values of n_x , the number of PGMRES (P_1 GMRES and P_2 GMRES) iteration is much lesser than as compared to MC-based GMRES and TV-based CG to reach to the required accuracy $tol = 1e - 5$. The results for subsequent fixed point iterations are also similar.
- (3) From Table 4, it is observed that the PSNR and SSIM by MC-based PGMRES method is almost same as compared with the ordinary MC-based GMRES method but much higher than TV-based CG method for all values of n_x . But PGMRES method is generating this better PSNR and SSIM in quite less iterations. For example, to achieve the better PSNR and SSIM, the P_1 GMRES method needs only 40 iterations and the P_2 GMRES method needs only 24 iterations for $n_x = 64$. But the GMRES method needs more than 150 iterations to get the same PSNR and SSIM. The TV-based CG method is also getting more than 150 iterations to get its lower PSNR and SSIM. Same is the case for other values of n_x . Which means that MC-based PGMRES method is faster than the MC-based GMRES and TV-based CG method for nontexture images. Here, the performance of the preconditioner P_2 is again more effective than the preconditioner P_1

Table 4. Comparison of CG, GMRES and PGMRES for Example 3.

| Mesh size h | Blurred PSNR | Method | Deblurred PSNR | Deblurred SSIM | Iterations |
|------------------|-----------------|-------------|-------------------|-------------------|------------|
| 1/64 | 24.4401 | CG | 53.4730 | 0.9486 | 150+ |
| | 24.4401 | GMRES | 59.6722 | 0.9518 | 150+ |
| | 24.4401 | P_1 GMRES | 60.3144 | 0.9597 | 40 |
| | 24.4401 | P_2 GMRES | 59.9128 | 0.9614 | 24 |
| 1/128 | 27.0862 | CG | 50.5325 | 0.9596 | 150+ |
| | 27.0862 | GMRES | 56.8443 | 0.9682 | 150+ |
| | 27.0862 | P_1 GMRES | 56.8449 | 0.9690 | 70 |
| | 27.0862 | P_2 GMRES | 56.7674 | 0.9689 | 38 |
| 1/256 | 28.9698 | CG | 46.2514 | 0.8439 | 150+ |
| | 28.9698 | GMRES | 49.2330 | 0.8751 | 150+ |
| | 28.9698 | P_1 GMRES | 49.2337 | 0.8752 | 92 |
| | 2896.98 | P_2 GMRES | 49.2335 | 08752 | 50 |

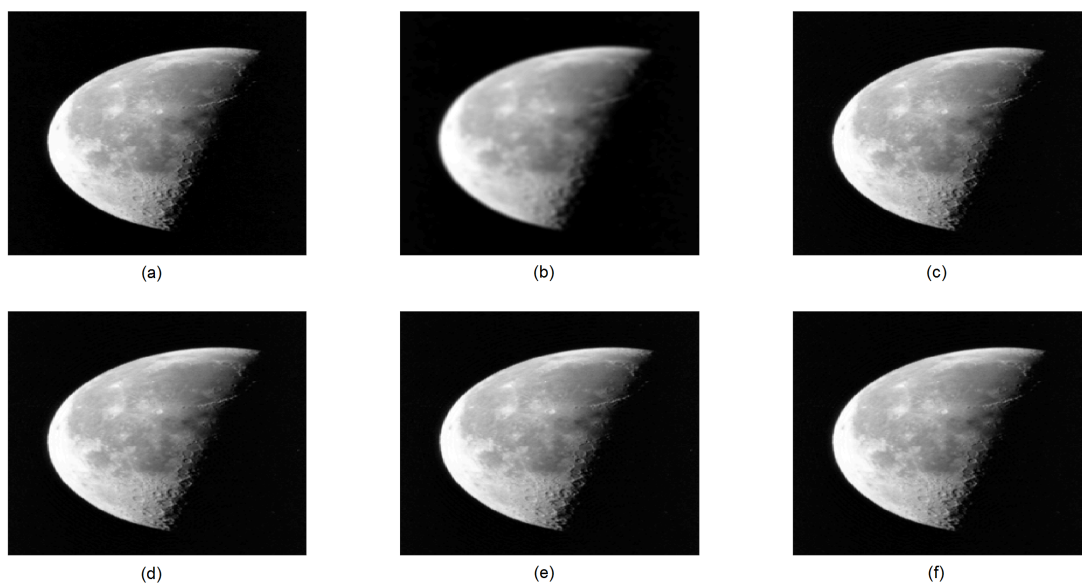


Figure 6. Moon image: (a) exact image; (b) blurry image; (c) deblurred image by CG; (d) deblurred image by GMRES; (e) deblurred image by P_1 GMRES; (f) deblurred image P_2 GMRES.

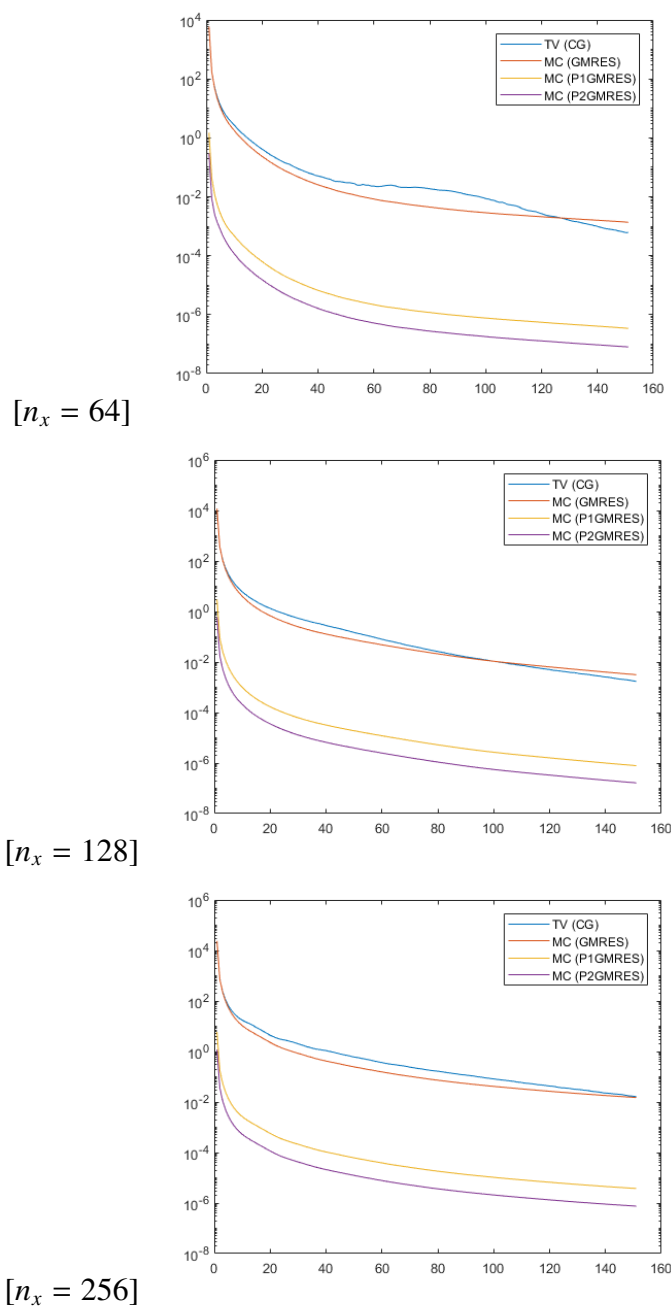


Figure 7. The CG, GMRES and PGMRES convergence for the solution to the linear system in Example 3 at fixed point iteration $m = 1$. The horizontal axis represents the number of iterations. The norm of the residual at each iteration is presenting on vertical axis.

Example 4. In this example, we have compared our MC-based algorithm with MC-based augmented lagrangian methods of (Zhu, Tai and Chan) [19, 20] and (Zhang, Deng, Shi, Wang and Yonggui) [17]. We have done the comparison for image denoising case (when blurring operator $\vec{K} = I$ identity operator) because in the given literature [17, 19, 20] augmented lagrangian method proposed only for image denoising problem. For simplicity, we have named augmented lagrangian methods in [19, 20]

and [17] by FFTALM, GSALM and LALM, respectively. For the comparison we have used Lena and Cameraman images. The different aspects of both images are presented in Figure 8 and 9. The size of each subfigure is 256×256 . The MC-based augmented lagrangian methods (FFTALM, GSALM and LALM) use certain sets of parameters which we have used according to [17, 19, 20]. For FFTALM, the parameters are $\alpha = 8e - 1, r_1 = 3e - 4, r_2 = 50, r_3 = 5e + 1, r_4 = 5e + 4$ and $\beta = 1$. For GSALM, the parameters are $\alpha = 8e - 1, r_1 = 4e - 3, r_2 = 50, r_3 = 1e + 3, r_4 = 5e + 4$ and $\beta = 1$. For LALM, the parameters are $\alpha = 8e - 1, r_1 = 4e - 3, r_2 = 50, r_3 = 1e + 3, r_4 = 5e + 4, \beta = 1, \delta_1 = 0.0038$ and $\delta_2 = 0.00012$. For our MC-based algorithm (GMRES and PGMRES) we have used $\alpha = 1e - 6, \beta = 1$ and $\gamma = 1000$. In both noisy images we have used random noise with mean ($\mu = 0$) and standard deviation ($\sigma = 10$). For the stopping criteria of a numerical methods we have used tolerance $tol = 9e - 4$ for Lena image and $tol = 8e - 4$ for Cameraman image.

Remarks

- (1) From the Figures 8 and 9 one can notice that all methods are generating same quality results.
- (2) From the Table 5, it is observed that the PSNR value of all methods is almost same for both images. But our methods (GMRES, P_1 GMRES and P_2 GMRES) are generating same PSNR in quite less CPU-Time as compared to all augmented lagrangian methods (FFTALM, GSALM and LALM). For example, the P_1 GMRES method is generating its PSNR for Lena image in just 6.528 seconds. But FFTALM, GSALM and LALM are generating their PSNR in 19.515, 11.406 and 9.078 seconds, respectively. Which means P_1 GMRES method is saving more than 60% of CPU-Time as compared to FFTALM, more than 50% of CPU-Time as compared to GSALM and more than 30% of CPU-Time as compared to LALM. Same is the case for P_2 GMRES method. So we can say, our proposed MC-based methods (GMRES, P_1 GMRES and P_2 GMRES) are faster than the MC-based augmented lagrangian methods (FFTALM, GSALM and LALM).



Figure 8. Lena image: (a) noisy image; (b) denoised image by FFTALM; (c) denoised image by GSALM; (d) denoised image by LALM; (e) denoised image by GMRES; (f) denoised image by P_1 GMRES; (g) denoised image by P_2 GMRES.

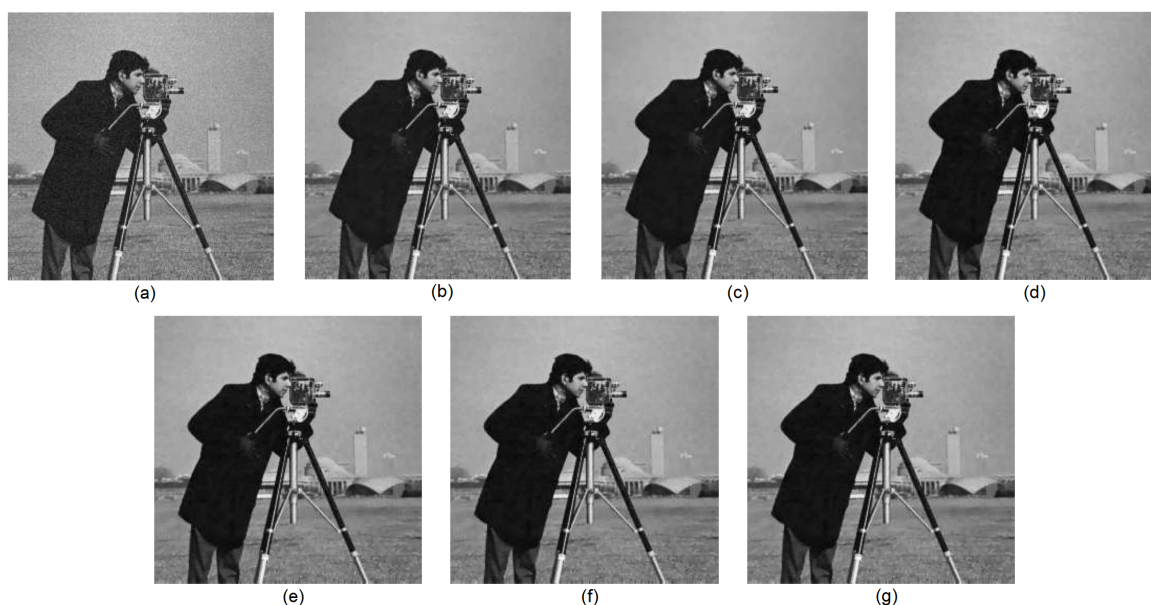


Figure 9. Cameraman image: (a) noisy image; (b) denoised image by FFTALM; (c) denoised image by GSALM; (d) denoised image by LALM; (e) denoised image by GMRES; (f) denoised image by P_1 GMRES; (g) denoised image by P_2 GMRES.

Table 5. Comparison of different methods for Example 4.

| Image | Noisy PSNR | Method | Denoised PSNR | CPU-Time |
|-----------|------------|-------------|---------------|----------|
| Lena | 28.154 | FFTALM | 32.970 | 19.515 |
| | 28.154 | GSALM | 33.103 | 11.406 |
| | 28.154 | LALM | 33.022 | 9.078 |
| | 28.154 | GMRES | 33.051 | 8.931 |
| | 28.154 | P_1 GMRES | 32.952 | 6.528 |
| | 28.154 | P_2 GMRES | 33.033 | 6.796 |
| Cameraman | 28.105 | FFTALM | 32.593 | 22.578 |
| | 28.105 | GSALM | 32.676 | 13.734 |
| | 28.105 | LALM | 32.422 | 10.390 |
| | 28.105 | GMRES | 32.547 | 10.439 |
| | 28.105 | P_1 GMRES | 32.109 | 7.159 |
| | 28.105 | P_2 GMRES | 32.454 | 7.985 |

5. Conclusions

A numerical algorithm (PGMRES) is presented to solve the primal form of mean curvature-based nonlinear image deblurring problem. Two new SPD preconditioner matrices (P_1 and P_2) are introduced. Four examples are tested by PGMRES using our new preconditioner matrices. Different kinds of images (Complicated, real, synthetic and nontexture) are tested by PGMRES using our new

preconditioner matrices. The convergence rates for solution to the linear system and the norm of the residual at each iteration are presented in each example. The comparison between our proposed MC-based methods (GMRES, P_1 GMRES and P_2 GMRES) and MC-based augmented lagrangian methods (FFTALM, GSALM and LALM) is also presented. Numerical experiments show the rapid convergence of PGMRES method using new preconditioners. Which means that PGMRES method is a robust and faster numerical method. It is also observed that the performance of the preconditioner P_2 is more effective than the preconditioner P_1 .

Acknowledgments

The authors would like to thank King Fahd University of Petroleum and Minerals (KFUPM) for its continuous supports. The authors also thank the referees for their very careful reading and valuable comments. This work is funded by KFUPM under Project #SB201012.

Conflict of interest

The authors declare that there is no conflict of interest.

References

1. R. Acar, C. R. Vogel, Analysis of bounded variation penalty methods for ill-posed problems, *Inverse Probl.*, **10** (1994), 1217–1229.
2. C. Brito-Loeza, K. Chen, Multigrid algorithm for high order denoising, *SIAM J. Imaging Sci.*, **3** (2010), 363–389.
3. C. Brito-Loeza, K. Chen, V. Uc-Cetina, Image denoising using the gaussian curvature of the image surface, *Numer. Meth. Part. D. E.*, **32** (2016), 1066–1089.
4. K. Chen, Introduction to variational image-processing models and applications, *Int. J. Comput. Math.*, **90** (2013), 1–8.
5. K. Chen, F. Fairag, A. Al-Mahdi, Preconditioning techniques for an image deblurring problem, *Numer. Linear Algebr.*, **23** (2016), 570–584.
6. F. Fairag, S. Ahmad, A two-level method for image deblurring problem, In: *2019 8th International Conference on Modeling Simulation and Applied Optimization (ICMSAO)*, 2019, 1–5.
7. F. Fairag, K. Chen, S. Ahmad, Analysis of the ccf method for mc-based image denoising problems, *Electron. T. Numer. Ana.*, **54** (2021), 108–127.
8. M. Myllykoski, R. Glowinski, T. Karkkainen, T. Rossi, A new augmented lagrangian approach for ℓ^1 -mean curvature image denoising, *SIAM J. Imaging Sci.*, **8** (2015), 95–125.
9. K. L. Riley, *Two-level preconditioners for regularized ill-posed problems*, Montana State University, 1999.
10. L. I. Rudin, S. Osher, E. Fatemi, Nonlinear total variation based noise removal algorithms, *Physica D*, **60** (1992), 259–268.

11. H. Rui, H. Pan, A block-centered finite difference method for the Darcy-Forchheimer model, *SIAM J. Numer. Anal.*, **5** (2012), 2612–2631.
12. L. Sun, K. Chen, A new iterative algorithm for mean curvature-based variational image denoising, *BIT*, **54** (2014), 523–553.
13. A. N. Tikhonov, Regularization of incorrectly posed problems, *Sov. Math. Dokl.*, **4** (1963), 1624–1627.
14. C. R. Vogel, M. E. Oman, Fast, robust total variation-based reconstruction of noisy, blurred images, *IEEE T. Image Process.*, **7** (1998), 813–824.
15. F. Yang, K. Chen, B. Yu, Homotopy method for a mean curvature-based denoising model, *Appl. Numer. Math.*, **62** (2012), 185–200.
16. F. Yang, K. Chen, B. Yu, D. Fang, A relaxed fixed point method for a mean curvature-based denoising model, *Optim. Method. Softw.*, **29** (2014), 274–285.
17. J. Zhang, C. Deng, Y. Shi, S. Wang, Y. Zhu, A fast linearised augmented lagrangian method for a mean curvature based model, *E. Asian J. Appl. Math.*, **8** (2018), 463–476.
18. W. Zhu, T. Chan, Image denoising using mean curvature of image surface, *SIAM J. Imaging Sci.*, **5** (2012), 1–32.
19. W. Zhu, X. C. Tai, T. Chan, Augmented lagrangian method for a mean curvature based image denoising model, *Inverse Probl. Imag.*, **7** (2013), 1409–1432.
20. W. Zhu, X. C. Tai, T. Chan, A fast algorithm for a mean curvature based image denoising model using augmented lagrangian method, In: *Efficient algorithms for global optimization methods in computer vision*, Springer, 2014, 104–118.



AIMS Press

©2021 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>)