Circulant preconditioners for mean curvature-based image deblurring problem

Shahbaz Ahmad and Faisal Fairag

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 produces a nonlinear ill-conditioned system which affects the convergence of the numerical algorithms such as Krylov subspace methods (generalized minimal residual etc.). To overcome this difficulty, in this paper, we present three new circulant preconditioners. An efficient algorithm is presented for the mean curvature-based image deblurring problem, which combines a fixed point iteration 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 circulant preconditioners.

Keywords
Image deblurring, mean curvature, ill-posed problem, numerical analysis, precondition matrix

Received 25 April 2021; accepted 8 October 2021

Introduction
In the last two decades, 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, blurred images. One of them, of course, is nonlinearity and the other is the solution of the large system which arises from the discretization after linearization. The main focus of this paper is to handle these two computational difficulties. The most well-known nonlinear variational image deblurring model is the total variation (TV) model. It has nice properties such as 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)-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 the staircase effect in the recovery of digital images. However, the discretization of the associated Euler–Lagrange equations produces a large nonlinear ill-conditioned system which affects the convergence of the numerical algorithms such as Krylov subspace methods (generalized minimal residual (GMRES), etc.). Furthermore, the Jacobian matrix of the MC-based nonlinear system is a block banded matrix with a large bandwidth. The MC-based regularization methods are effective, but due to the high nonlinearity and ill-conditioned system, a robust and fast numerical solution is a crucial issue. To overcome these difficulties, in this paper, we introduce three new circulant symmetric positive definite (SPD) preconditioners. So instead of applying the ordinary GMRES method (without preconditioner), we use preconditioned GMRES (PGMRES) method (with new preconditioners) for the solution of the system. A fast convergence has been shown in the numerical results by using the proposed new preconditioners.

The contributions of the study include the following: (i) our work presents an efficient algorithm for an MC-based image deblurring problem, which combines a fixed point iteration (FPI) with new circulant preconditioned matrices

1Department of Mathematics, KFUPM, Saudi Arabia

Corresponding author:
Department of Mathematics, KFUPM, Saudi Arabia.
Email: shahbazahmad@kfupm.edu.sa

Creative Commons Non Commercial CC BY-NC: This article is distributed under the terms of the Creative Commons Attribution-NonCommercial 4.0 License (https://creativecommons.org/licenses/by-nc/4.0/) which permits non-commercial use, reproduction and distribution of the work without further permission provided the original work is attributed as specified on the SAGE and Open Access page (https://us.sagepub.com/en-us/nam/open-access-at-sage).
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 MC regularization functional. The paper is organized into different sections. The first section includes an introduction while the second section includes a problem description of an image deblurring model. In the third section, we present a nonlinear system of first-order equations and cell discretization. In the fourth section, we introduced our proposed circulant preconditioners for the PGMRES method. The properties and eigenvalues analysis of the proposed preconditioned matrices are discussed in the fourth section. The numerical experiments are also in the fourth section. The conclusions about the proposed PGMRES method are discussed in the last section of the paper.

Problem description

The focus of the paper is on the 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:

$$z = \tilde{K}u + e \quad (1)$$

where $e$ is the noise function. The noise can be Gaussian noise, salt and pepper noise, Brownian noise, etc. The $\tilde{K}$ is the blurring operator, which is a Fredholm-integral operator of the first kind:

$$\tilde{K}u(x) = \int \Omega k(x, y)u(y) \, dy, \quad x \in \Omega$$

having translation invariance property, that is, $k(x, y) = k(x - y)$. The problem (1) becomes ill-posed because $\tilde{K}$ is a compact operator. Let $\Omega$ be a square in $\mathbb{R}^2$, $u \in \Omega$ is an image intensity function. The $x = (x, y)$ defines the position in $\Omega$. Let $||x|| = \sqrt{x^2 + y^2}$ is an Euclidean norm and $||.||$ is $L_2(\Omega)$ norm. Equation (1) is an inverse problem. The recovering of $u$ from $z$ makes (1) an unstable problem. To make it stable, one remedy is to use the MC regularization functional:

$$J(u) = \int \Omega (\kappa u)^2 \, dx = \int \Omega \left( \nabla \frac{\nabla u}{|\nabla u|} \right)^2 \, dx$$

Then problem (1) takes the form, find $u$ that minimize

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

where $\alpha > 0$ is a regularization parameter. The well-posedness of problem (2) for a particular case (synthetic image denoising problem) is explained in Zhu and Chan.17 Then, the Euler-Lagrange equations of (2) are as follows:

$$\nabla \cdot \left( \frac{\nabla \kappa}{\sqrt{||\nabla u||^2 + \beta^2}} - \frac{\nabla \kappa \cdot \nabla u}{\sqrt{||\nabla u||^2 + \beta^2}^3} \nabla u \right) = 0 \quad \text{in} \ \Omega$$

(3)

$$\frac{\partial u}{\partial n} = 0 \quad \text{in} \ \partial \Omega$$

(4)

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

(5)

where $\tilde{K}^*$ denotes the adjoint operator of $\tilde{K}$ and $\beta > 0$ is used to avoid non-differentiability at zero. Equation (3) is a nonlinear fourth-order differential equation.

The MC-based model not only preserves edges but also removes the 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 MC functional. We have treated this difficulty by reducing the nonlinear fourth-order Euler-Lagrange equation into a system of first-order equations.

The first-order nonlinear system

Equation (3) can be expressed as a first-order nonlinear system:

$$\tilde{K}^* \tilde{K}u + \alpha \nabla \cdot \tilde{p} - \alpha \nabla \cdot \tilde{t} = \tilde{K}^* z \quad (6)$$

$$-w + \nabla \cdot \tilde{v} = 0 \quad (7)$$

$$\sqrt{||\nabla u||^2 + \beta^2} \tilde{v} - \nabla u = 0 \quad (8)$$

$$\sqrt{||\nabla u||^2 + \beta^2} \tilde{p} - \nabla w = 0 \quad (9)$$

$$\sqrt{||\nabla u||^2 + \beta^2} \tilde{t} - (\nabla w \cdot \tilde{v}) \tilde{v} = 0 \quad (10)$$

where

$$\tilde{v} = \frac{\nabla u}{\sqrt{||\nabla u||^2 + \beta^2}}, \quad w = \nabla \cdot \tilde{v}, \quad \tilde{p} = \frac{\nabla w}{\sqrt{||\nabla u||^2 + \beta^2}}$$

and

$$\tilde{t} = \frac{(\nabla w \cdot \tilde{v}) \tilde{v}}{\sqrt{||\nabla u||^2 + \beta^2}}$$

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

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

\[
 \delta_x : 0 = x_1 \leq x_2 \leq x_3 \leq \cdots \leq x_{n_1} \leq x_{n_1+1} = 1
\]

\[
 \delta_y : 0 = y_1 \leq y_2 \leq y_3 \leq \cdots \leq y_{n_2} \leq y_{n_2+1} = 1
\]

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

\[
x_i = \left( i - \frac{1}{2} \right) h \quad i = 1, 2, 3, \ldots, n_x
\]

\[
y_j = \left( j - \frac{1}{2} \right) h \quad j = 1, 2, 3, \ldots, n_y
\]

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

\[
x_{i\frac{1}{2}} = x_i \pm \frac{h}{2} \quad i = 1, 2, 3, \ldots, n_x
\]

\[
y_{j\frac{1}{2}} = y_j \pm \frac{h}{2} \quad j = 1, 2, 3, \ldots, n_y
\]

The set

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

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

\[
\chi_i(x) = \begin{cases} 1 & x \in (x_i - \frac{1}{2}, x_i + \frac{1}{2}) \\ 0 & \text{otherwise} \end{cases}
\]

\[
\chi_j(y) = \begin{cases} 1 & y \in (y_j - \frac{1}{2}, y_j + \frac{1}{2}) \\ 0 & \text{otherwise} \end{cases}
\]

And

\[
\phi_i \left( x_i + \frac{1}{2} \right) = \delta_{il}
\]

\[
\phi_j \left( y_j + \frac{1}{2} \right) = \delta_{jk}
\]

An approximation of \( u \) and \( w \) is as follows:

\[
u(x, y) \approx \mathcal{U}(x, y) = \sum_{i=1}^{n_x} \sum_{j=1}^{n_y} u_{ij} \chi_i(x) \chi_j(y)
\]

and

\[
w(x, y) \approx \mathcal{W}(x, y) = \sum_{i=1}^{n_x} \sum_{j=1}^{n_y} w_{ij} \chi_i(x) \chi_j(y)
\]

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

\[
z(x, y) \approx \mathcal{Z}(x, y) = \sum_{i=1}^{n_x} \sum_{j=1}^{n_y} z_{ij} \chi_i(x) \chi_j(y)
\]

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

\[
(Ku)(x_i, y_j) \approx [K_h U]_{ij}
\]

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

\[
\overrightarrow{V}^x(x, y) = \sum_{i=1}^{n_x-1} \sum_{j=1}^{n_y} \chi_i(x) \chi_j(y)
\]

\[
\overrightarrow{V}^y(x, y) = \sum_{i=1}^{n_x} \sum_{j=1}^{n_y-1} \chi_i(x) \chi_j(y)
\]

respectively. \( \mathbf{V} = [\mathbf{V}^x \mathbf{V}^y] \) denotes the discretization of \( \mathbf{v} \). Similarly, the approximation of the components of \( \mathbf{p} \) and \( \mathbf{t} \) is

\[
\overrightarrow{P}^x(x, y) = \sum_{i=1}^{n_x-1} \sum_{j=1}^{n_y} \chi_i(x) \chi_j(y)
\]

\[
\overrightarrow{P}^y(x, y) = \sum_{i=1}^{n_x} \sum_{j=1}^{n_y-1} \chi_i(x) \chi_j(y)
\]

and

\[
\overrightarrow{T}^x(x, y) = \sum_{i=1}^{n_x-1} \sum_{j=1}^{n_y} \chi_i(x) \chi_j(y)
\]

\[
\overrightarrow{T}^y(x, y) = \sum_{i=1}^{n_x} \sum_{j=1}^{n_y-1} \chi_i(x) \chi_j(y)
\]

respectively. The \( \mathbf{P} = [\mathbf{P}^x \mathbf{P}^y] \) and \( \mathbf{T} = [\mathbf{T}^x \mathbf{T}^y] \) denote the discretization of the vectors \( \mathbf{p} \) and \( \mathbf{t} \), respectively.

The cell-centered finite difference method

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

\[
U = [U_{11} \cdots U_{n_x n_y}], \quad W = [W_{11} \cdots W_{n_x n_y}], \quad V
\]

\[
P = [P_{11}^x \cdots P_{n_x-1 n_y}^x \cdots P_{n_x-1 n_y}^y], \quad \mathbf{v}
\]

\[
T = [T_{11}^x \cdots T_{n_x-1 n_y}^x \cdots T_{n_x-1 n_y}^y]
\]

and

Now by applying the CCFD method to (6) to (10), we obtain the following system:
\[ K^*K_h U - \alpha A_h W + aB_h^* P - aB_h^* T = K_h^* Z \]  \hspace{1cm} (11)

\[-l_h W + B_h^* V = O \]  \hspace{1cm} (12)

\[ D_h V + B_h U = O \]  \hspace{1cm} (13)

\[ D_h P + B_h W = O \]  \hspace{1cm} (14)

\[ D_h T - C_h V = O \]  \hspace{1cm} (15)

where the midpoint quadrature rule is used for the integral term. The matrices \( K_h \), \( A_h \), and \( l_h \) are 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:

\[ \begin{bmatrix} K_h^*K_h & -A_h & O & aB_h^* & -aB_h^* \\ O & -l_h & B_h^* & O & O \\ B_h & O & D_h & O & O \\ O & B_h & O & D_h & O \\ O & O & -C_h & O & D_h \end{bmatrix} \begin{bmatrix} U \\ W \\ V \\ P \\ T \end{bmatrix} = \begin{bmatrix} K_h^*Z \\ O \\ O \\ O \end{bmatrix} \]

The matrix \( K_h \) is block Toeplitz with Toeplitz blocks and \( K_h^*K_h \) is SPD. The matrix \( A_h \) is a diagonal matrix having the following structure:

\[ A_h = \frac{2}{\beta h}(A_1 + A_2) \]

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

\[ A_1 = \tilde{l} \otimes E \quad \text{and} \quad A_2 = E \otimes \tilde{l} \]

where \( \otimes \) is a tensor product. The size of the identity matrix \( \tilde{l} \) is \( n_x \times n_x \). The matrix \( E \) 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

\[ B_1 = F \otimes \tilde{l} \quad \text{and} \quad B_2 = \tilde{l} \otimes F. \]

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

\[ C_h = \begin{bmatrix} C^* & 0 \\ 0 & C^* \end{bmatrix} \]

is a diagonal matrix and its entries are obtained by the discretization of the expression \( \nabla \cdot w \cdot \nabla \). The matrix \( C^* \) is of size \( (n_x - 1) \times n_x \), and the matrix \( C^* \) 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 \( \sqrt{1 + \|
abla u\|^2 + \beta^2} \). The matrix \( D_h \) has the following structure:

\[ D_h = \begin{bmatrix} D^x & 0 \\ 0 & D^x \end{bmatrix} \]

where \( D^x \) is of size \( (n_x - 1) \times n_x \) and \( D^x \) 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 (11) to (15), then we have the following primal system:

\[ (K_h^*K_h + \alpha L_h(U))U = K_h^*Z \]  \hspace{1cm} (16)

where

\[ L_h = (B_h^*D_h^{-1}B_h)^2 + A_h(B_h^*D_h^{-1}B_h) \]

\[ + B_h^*D_h^{-1}C_hD_h^{-1}B_h \]  \hspace{1cm} (17)

The \( L_h \) is a 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 MC-based nonlinear image minimization problems. 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 MC regularization functional in the governing equation of the models makes these highly nonlinear systems harder for calculation. MC is a very much computationally expensive, that is why most of the existing methods perform quite poorly. To make a robust numerical method for the MC-based nonlinear image deblurring problem, now we present a new preconditioned numerical method.
Numerical implementation

Here we introduce the algorithms to solve the MC-based nonlinear system (16). First, we apply a discrete version of the FPI to (16) to handle the nonlinearity of MC. The approach taken here is called the “lagged diffusivity” scheme. Its rate is just linear but in practice, it has a quite rapid convergence. Furthermore, this scheme does not depend on the initial guess to converge globally. This is why globalization is not an issue for this scheme. So by FPI, we have the following linear system:

\[ (K_h^* K_h + \alpha L_h(U^m)) U^{m+1} = K_h^* Z \]  \hspace{1cm} (18)

Properties

Before proceeding further, we discuss some important properties of our system (18).

1. The Hessian matrix \( K_h^* K_h + \alpha L_h \) is extremely large for practical applications. When \( \alpha \) is small, the Hessian matrix becomes quite ill-conditioned. This happens because the eigenvalues of the blurring operator \( K \) cluster as zero.4
2. The first term \( K_h^* K_h \) in the Hessian matrix is SPD. Although, \( K_h^* K_h \) is full but the blurring operator \( K \) has a translation invariant property, which permit the use of fast Fourier transformation (FFT) to evaluate \( K_h^* K_h \mu \) in \( O(n \log n) \) operations.4
3. The second term \( L_h \) in the Hessian matrix is sparse but not symmetric. The \( L_h \) (16) consists of three terms. The first and the last term in \( L_h \) are symmetric positive semidefinite but the middle term is not symmetric. Hence, system (18) is not SPD.

Preconditioned matrices

According to the properties of our system (18), mentioned above, the GMRES method is suitable for the solution of system (18). Due to an ill-conditioned system, GMRES can be quite slow to convergence. So we use the PGMRES method. For an effective solution, the preconditioning matrix must be SPD. Here we introduce the following three SPD circulant preconditioned matrices. The first preconditioning matrix \( P_1 \) is

\[ P_1 = (\gamma K_h^* K_h + \alpha I_h)^{1/2} (\gamma L_h + I_h)(\gamma K_h^* K_h + \alpha I_h)^{1/2} \]  \hspace{1cm} (19)

where \( \gamma \) is the positive parameter. The \( K_h^* K_h \) is a circulant approximation of matrix \( K_h \) and \( I_h \) is an identity matrix. The second preconditioning matrix \( P_2 \)

\[ P_2 = (\gamma K_h^* K_h + \alpha I_h)^{1/2} (\gamma L_h + I_h)(\gamma K_h^* K_h + \alpha I_h)^{1/2} \]  \hspace{1cm} (20)

Algorithm 4.1. The PGMRES method

1: On mesh \( \Omega_h \), Initial iteration \( U^0 \),
2: \( \text{for } m = 1; \max \text{ } \longrightarrow \)
   \[ A^m = K_h^* K_h + \alpha L_h(U^m), \]
   \[ b^m = K_h^* Z, \]
3: Use PGMRES to solve \( A^m U^{m+1} = b^m \),
   with preconditioned matrix \( P = P_1, P_2, P_3 \).
   end

where the matrix

\[ L_h^{TV} = (B_h^1 D_h^{-1} B_h)^2 + B_h^2 D_h^{-1} C_h D_h^{-1} B_h \]  \hspace{1cm} (21)

is the SPD part (the first and the third term) of \( L_h \) matrix (18). The third preconditioning matrix \( P_3 \) is

\[ P_3 = (\gamma K_h^* K_h + \alpha I_h)^{1/2} (\gamma L_h^{TV} + I_h)(\gamma K_h^* K_h + \alpha I_h)^{1/2} \]  \hspace{1cm} (22)

where the matrix

\[ L_h^{TV} = B_h^1 D_h^{-1} B_h, \]  \hspace{1cm} (23)

is the SPD matrix that lies in the first term of \( L_h \) matrix (18). In fact, \( L_h^{TV} \) arise by the discretization of the TV-based image deblurring problem.4

The preconditioned matrices \( P_1, P_2, \) and \( P_3 \) are “product preconditioners” based on the analogy of “operator splitting” techniques.23,4 While applying PGMRES to (18), the inversion of preconditioner matrices \( (P_1, P_2, \) and \( P_3 \)) will be required. The inversion of the preconditioning matrices \( (P_1, P_2, \) and \( P_3 \)) requires the inversion of the middle terms of (17), (20), and (20), respectively. The middle terms of \( P_1, P_2, \) and \( P_3 \) have the following forms: \( \gamma L_h + I_h, \gamma L_h^{TV} + I_h, \) and \( \gamma L_h^{TV} + I_h, \) respectively. Since these middle terms are sparse matrices, so inversion can be done easily. The inversion of the preconditioning matrices \( (P_1, P_2, \) and \( P_3 \)) also require the inversion of the matrix \( (\gamma K_h^* K_h + \alpha I_h)^{1/2} \). For this, we need only \( O(n \log n) \) floating point operations using FFTs. The detail is given in Vogel and Oman.4 The PGMRES method is summarized in Algorithm 4.1:

Eigenvalues

Now, let the eigenvalues of \( K_h^* K_h + \alpha L_h \) be \( \lambda_i^K \) and \( \lambda_i^L \), respectively, such that \( \lambda_i^K \downarrow 0 \) and \( \lambda_i^L \uparrow \infty \). So the eigenvalues of \( P_1^{-1} \tilde{A} \) are as follows:

\[ \theta_i^1 = \frac{\lambda_i^K + \alpha \lambda_i^L}{(\gamma \lambda_i^K + \alpha)(\gamma \lambda_i^L + 1)} \]  \hspace{1cm} (24)

where \( \tilde{A} = K_h^* K_h + \alpha L_h \) is the Hessian matrix of system (18). Clearly, if one can choose small \( \gamma \) such that \( \alpha \leq \gamma \frac{\lambda_{i-1}^L}{\lambda_i^K} \), then \( \theta_i^1 \to 1 \) as \( i \to \infty \). Now consider the
Clearly, for $\alpha \leq \gamma \leq (\lambda_i^L - 1)/\lambda_i^L$, these are

$$
\theta_i^1 = \frac{\lambda_i^E + \alpha \lambda_i^L}{(\gamma \lambda_i^L + \alpha)(\gamma \lambda_i^L + 1)} \quad \text{and} \quad 
\theta_i^2 = \frac{\lambda_i^E + \alpha \lambda_i^L}{(\gamma \lambda_i^L + \alpha)(\gamma \lambda_i^L + 1)}
$$

respectively. Here, $\lambda_i^{TV}$ and $\lambda_i^{L^2}$ are eigenvalues of $L_i^{TV}$ and $L_i^{L^2}$, respectively. One can notice that $\lambda_i^{TV} \leq \lambda_i^{L^2} \leq \lambda_i^L$. Then

$$
\theta_i^2 \geq \frac{\lambda_i^E + \alpha \lambda_i^L}{(\gamma \lambda_i^L + \alpha)(\gamma \lambda_i^L + 1)} \quad \text{and} \quad 
\theta_i^3 \geq \frac{\lambda_i^E + \alpha \lambda_i^L}{(\gamma \lambda_i^L + \alpha)(\gamma \lambda_i^L + 1)}
$$

Clearly, for $\alpha \leq \gamma \leq \lambda_i^L - 1$,

$$
\theta_i^2, \theta_i^3 \to 1 \quad \text{as} \quad i \to \infty
$$

Hence, for $\alpha \leq \gamma \leq [(\lambda_i^L - 1)/\lambda_i^L] \leq 1$, $P_1^{-1} \Delta$, $P_2^{-1} \Delta$ and $P_3^{-1} \Delta$ have more favorable spectrum as compared to the Hessian matrix $\Delta$, so PGMR (P1 GMRES, P2 GMRES and P3 GMRES) converges more rapidly than ordinary GMRES. This can also be shown in the numerical examples.

Now we present four numerical examples for the MC-based image deblurring problem. We have used different values of $n$, so the resulting system has $n^2$ unknowns. In all examples, the value of the parameters $\alpha$ and $\beta$ is used according to the literature. \cite{24,17,18} For numerical computations, MATLAB software is used to obtain the numerical results on Intel(R) Core(TM) i7-4510U with CPU@2.00 GHz and 2.60 GHz. To measure the quality of the deblurred images, we have used peak signal-to-noise ratio (PSNR), signal-to-noise ratio (SNR), and structured similarity index measure (SSIM).

**Example 1:** In this example, we have used Lena image. This is a complicated image, because it has a small-scale texture part (hat) and a large-scale cartoon part (face). The different aspects of the Lena image are presented in Figure 1. The size of each subfigure is $512 \times 512$. These are (a) exact image, (b) blurry image, (c) deblurred image by GMRES, (d) deblurred image by $P_1$ GMRES with $\gamma = 1e - 2$, (e) deblurred image by $P_2$ GMRES with $\gamma = 1e - 6$, and (f) deblurred image by $P_3$ GMRES with $\gamma = 1e - 6$.

![Figure 1](image.png)

**Remarks:**

1. One can notice that Figure 1(d) to (f) is having much better quality as compared with Figure 1(c). This means the PGMR (P1 GMRES, P2 GMRES and P3 GMRES) method is generating better quality results by using the ordinary GMRES method without preconditioning.

2. From Figure 2, one can clearly observe the effectiveness of preconditioning. Here, the result was presented using an FPI count $m = 1$. The number of PGMR iteration is much lesser than as compared to GMRES (without preconditioning) to reach the required accuracy tol $= 1e - 7$. The later FPIs are also having similar results.

3. From Figure 2, this can also be observed that PGMR is getting rapid convergence when the value of $\gamma$ is close to the value of $\alpha = 1e - 8$ for all preconditioners ($P_1$, $P_2$, and $P_3$).

4. From Table 1 and Figure 2, one can observe that the PSNR and SSIM by using the PGMR method is much better as compared with the ordinary GMRES method. But the PGMR method is producing better PSNR and SSIM in quite less iterations.

5. The $P_1$ GMRES method achieves its best PSNR and SSIM with $\gamma = 1e - 2$. The $P_2$ GMRES method needs only 11 iterations with $\gamma = 1e - 6$ to get his best PSNR and SSIM. Similar is the case for $P_3$ GMRES. This also justify the selection of the best $\gamma$ that is, $\alpha \leq \gamma \leq 1$, which we have observed in the
Figure 2. GMRES and PGMRES convergence at an FPI $m = 1$ for Example 1. The norm of the residual at each iteration is present on the vertical axis. The blue line represents GMRES iterations. (a) $P_1$ GMRES iteration against different values of $\gamma$ (b) $P_2$ GMRES iteration against different values of $\gamma$ and (c) $P_3$ GMRES iteration against different values of $\gamma$. GMRES: generalized minimal residual; PGMRES: preconditioned GMRES; FPI: fixed point iteration.
eigenvalue analysis (section “Eigenvalues”). Although, the value of $\gamma$ smaller than $\alpha$ further reduces the number of iterations but it also decreases the deblurred quality (PSNR and SSIM).

6. It is also observed that although the $P_1$ GMRES method achieves little higher PSNR and SSIM as compared with both $P_2$ GMRES and $P_3$ GMRES but $P_1$ GMRES needs much more iterations than $P_2$ GMRES and $P_3$ GMRES. So, in this example, the performance of preconditioners $P_2$ and $P_3$ is more effective than the preconditioner $P_1$.

Example 2: Here we have used a nontexture Peppers image. The different aspects of the Peppers image are presented in Figure 3. The size of each subfigure is $256 \times 256$. These are (a) exact image, (b) blurry image, (c) deblurred image by GMRES, (d) deblurred image by $P_1$ GMRES with $\gamma = 1e-4$, (e) deblurred image by $P_2$ GMRES with $\gamma = 1e-5$, and (f) deblurred image $P_3$ GMRES with $\gamma = 1e-5$. For numerical calculations, we have used the $ke_{-gen}(N/4, 300, 3)$ kernel. Here, $\alpha = 1e-6$ and $\beta = 0.1$. For comparison, we have used three different values of $n_x$. These are 64, 128, and 256. For the stopping criteria of numerical methods, we have used tolerance $tol = 1e-7$.

Remarks:
1. Figure 3(c) to (f) is almost similar, so all methods are generating same quality results.
2. From Figure 4, one can clearly notice the effectiveness of preconditioning. For all values of $n_x$, the number of PGREMS ($P_1$ GMRES, $P_2$ GMRES, and $P_3$ GMRES) iteration is much lesser than as compared to GMRES to reach the required accuracy $tol = 1e-7$. The later FPIs are also having similar results.
3. From Table 2, it is observed that the PSNR and SSIM by the PGREMS method are almost the same as compared with the ordinary GMRES method for all values of $n_x$. But the PGREMS method is generating this PSNR and

| Blurred PSNR | Blurred SSIM | Method | $\gamma$ | Deblurred PSNR | Deblurred SSIM | Iterations |
|--------------|--------------|--------|---------|----------------|----------------|------------|
| 24.9488      | 0.7902       | GMRES  |         | 32.1210        | 0.9795         | 50+        |
| 24.9488      | 0.7902       | $P_1$ GMRES | $1e-2$ | 44.5108        | 0.9631         | 50+        |
|              |              |        | $1e-4$  | 41.4644        | 0.9825         | 50          |
|              |              |        | $1e-6$  | 42.0135        | 0.9841         | 11          |
|              |              |        | $1e-8$  | 35.5159        | 0.9819         | 5           |
|              |              |        | $1e-10$ | 32.1702        | 0.9796         | 4           |
| 24.9488      | 0.7902       | $P_2$ GMRES | $1e-2$ | 41.6700        | 0.9832         | 50+        |
|              |              |        | $1e-4$  | 41.9035        | 0.9830         | 50          |
|              |              |        | $1e-6$  | 42.0184        | 0.9841         | 11          |
|              |              |        | $1e-8$  | 35.5160        | 0.9819         | 5           |
|              |              |        | $1e-10$ | 32.1702        | 0.9796         | 4           |
| 24.9488      | 0.7902       | $P_3$ GMRES | $1e-2$ | 41.8196        | 0.9830         | 50+        |
|              |              |        | $1e-4$  | 41.9048        | 0.9830         | 50          |
|              |              |        | $1e-6$  | 42.0184        | 0.9841         | 11          |
|              |              |        | $1e-8$  | 35.5160        | 0.9819         | 5           |
|              |              |        | $1e-10$ | 32.1702        | 0.9796         | 4           |

GMRES: generalized minimal residual; PGREMS: preconditioned GMRES; PSNR: peak signal-to-noise ratio; SSIM: structured similarity index measure.

Figure 3. Peppers image: (a) exact image, (b) blurry image, (c) deblurred image by generalized minimal residual (GMRES), (d) deblurred image by $P_1$ GMRES with $\gamma = 1e-4$, (e) deblurred image by $P_2$ GMRES with $\gamma = 1e-5$, and (f) deblurred image $P_3$ GMRES with $\gamma = 1e-5$. 
Figure 4. The GMRES and PGMRES convergence at $m = 1$ for Example 2. The norm of the residual at each iteration is presented on vertical axis. The blue line represents GMRES iterations, the orange line represents $P_1$ GMRES iterations, the yellow line represents $P_2$ GMRES iteration, and the purple line represents $P_3$ GMRES iteration. (a) $n_x = 64$, (b) $n_x = 128$ and (c) $n_x = 256$. GMRES: generalized minimal residual; PGMRES: preconditioned GMRES; FPI: fixed point iteration.
SSIM in quite less iterations. To achieve the PSNR and SSIM, the PGMRES ($P_1$ GMRES, $P_2$ GMRES and $P_3$ GMRES) method needs only 10 iterations for $n_x = 64$. But the GMRES method needs 100 plus iterations to get the same PSNR and SSIM. The same is the case for other values of $n_x$. Which means that the PGMRES method is faster than the GMRES for nontexture images.  

4. From Table 2 and Figure 4, it is observed that the performance of all the preconditioners ($P_1$, $P_2$, and $P_3$) is the same.

Example 3: In this example, we have used a Goldhills image. This is a real and synthetic image. Here we have compared our MC-based algorithm with TV-based algorithm. Since the TV-based model generates an SPD matrix system, so for the solution we have used the conjugate gradient (CG) method. The different aspects of the Goldhills image are presented in Figure 5. The size of each subfigure is 512 x 512. These are (a) blurry image, (b) deblurred image by CG, (c) deblurred image by GMRES, (d) deblurred image by $P_1$ GMRES, (e) deblurred image by $P_2$ GMRES, and (f) deblurred image $P_3$ GMRES. For numerical calculations, we have used the $k_e..g$en($N/2, 300, 2$) kernel. For the TV-based method, we have used $\alpha = 5e - 4$ and $\beta = 1$ according to. For MC-based method we have used $\alpha = 1e - 4$ and $\beta = 1$ and $\gamma = 1e - 3$. For the stopping criteria of a numerical methods we have used tolerance $tol = 1e - 2$.

Remarks:

1. From Table 3, it is observed that the PSNR and SSIM by MC-based (GMRES and PGMRES) methods are a little higher than the TV-based CG method. The same comparison can be observed from Figures 5(b) to (f). So MC-based (GMRES and PGMRES) methods are generating better quality results.

![Relative Residual Norms](image)

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

| Mesh size | Blurred PSNR | Blurred SSIM | Method   | Deblurred PSNR | Deblurred SSIM | Iterations |
|-----------|-------------|-------------|----------|----------------|----------------|------------|
| $1/64$    | 16.7522     | 0.6459      | GMRES    | 35.1023        | 0.9905         | 100+       |
|           | 16.7522     | 0.6459      | $P_1$ GMRES | 35.1023        | 0.9905         | 10         |
|           | 16.7522     | 0.6459      | $P_2$ GMRES | 35.1024        | 0.9905         | 10         |
|           | 16.7522     | 0.6459      | $P_3$ GMRES | 35.1024        | 0.9905         | 10         |
| $1/128$   | 18.3185     | 0.6775      | GMRES    | 36.1033        | 0.9840         | 200+       |
|           | 18.3185     | 0.6775      | $P_1$ GMRES | 36.0281        | 0.9838         | 25         |
|           | 18.3185     | 0.6775      | $P_2$ GMRES | 36.0292        | 0.9838         | 25         |
|           | 18.3185     | 0.6775      | $P_3$ GMRES | 36.0262        | 0.9838         | 25         |
| $1/256$   | 19.5651     | 0.7013      | GMRES    | 33.2606        | 0.9616         | 200+       |
|           | 19.5651     | 0.7013      | $P_1$ GMRES | 33.2261        | 0.9613         | 40         |
|           | 19.5651     | 0.7013      | $P_2$ GMRES | 33.0346        | 0.9587         | 40         |
|           | 19.5651     | 0.7013      | $P_3$ GMRES | 33.0314        | 0.9587         | 40         |

CG: conjugate gradient; GMRES: generalized minimal residual; PGMRES: preconditioned GMRES; PSNR: peak signal-to-noise ratio; SSIM: structured similarity index measure.

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

|          | CG          | GMRES       | $P_1$ GMRES | $P_2$ GMRES | $P_3$ GMRES |
|----------|-------------|-------------|-------------|-------------|-------------|
| Blurred PSNR | 23.8381     | 23.8381     | 23.8381     | 23.8381     | 23.8381     |
| Blurred SSIM | 0.3316      | 0.3316      | 0.3316      | 0.3316      | 0.3316      |
| Deblurred PSNR | 34.1331     | 34.5917     | 34.1467     | 34.0028     | 34.5874     |
| Deblurred SSIM | 0.8680      | 0.8758      | 0.8639      | 0.8524      | 0.8758      |
| Iterations | 100+        | 100+        | 20          | 40          | 10          |

CG: conjugate gradient; GMRES: generalized minimal residual; PGMRES: preconditioned GMRES; PSNR: peak signal-to-noise ratio; SSIM: structured similarity index measure.

![Goldhills image](image)
Figure 6. The CG, GMRES, and PGMRES convergence at an FPI $m = 1$ for Example 3. The norm of the residual at each iteration is presented on the vertical axis. The blue line represents CG iterations, the orange line represents GMRES iterations, the yellow line represents $P_1$ GMRES iteration, the purple line represents $P_2$ GMRES iteration, and the green line represents $P_3$ GMRES iteration. CG: conjugate gradient; GMRES: generalized minimal residual; PGMRES: preconditioned GMRES; FPI: fixed point iteration.
2. From Figure 6, 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 $\text{tol} = 1e^{-2}$. The later FPIs are also having similar results.

3. From Table 3, it is observed that the PSNR and SSIM by MC-based PGMRES method are almost the same as compared with the ordinary MC-based GMRES method. But PGMRES method is generating the same PSNR and SSIM in quite less iterations. To achieve the same PSNR and SSIM, the $P_1$ GMRES method needs only 20 iterations, the $P_2$ GMRES method needs only 40 iterations and the $P_3$ GMRES method needs only 10 iterations. But the GMRES method needs 100 plus iterations to get the same PSNR and SSIM. The TV-based CG method is also getting 100 plus iterations to get its PSNR and SSIM. Which means that the 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 $P_3$ is more effective than the preconditioners $P_1$ and $P_2$.

**Example 4:** In this example we have applied our MC-based (GMRES and PGMRES) algorithms on a blurry image with a high level of Gaussian noise (mean $= 0.1$, variance $= 0.04$). We have also presented a comparison of our algorithms with the TV-based (CG) algorithm. For this, we have used a complicated Cameraman image. The different aspects of the Cameraman image are presented in Figure 7. The size of each subfigure is $512 \times 512$. These are (a) noisy image, (b) denoised image by CG, (c) denoised image by GMRES, (d) denoised image by $P_1$ GMRES, (e) denoised image by $P_2$ GMRES, and (f) denoised image $P_3$ GMRES. For numerical calculations, we have used the $ke_{\text{gen}}(N, 300, 1)$ kernel. For the TV-based method, we have used $\alpha = 1e - 2$ and $\beta = 1$ according to Vogal and Oman. For the MC-based method, we have used $\alpha = 1e - 2$, $\beta = 1$, and $\gamma = 1e - 1$. For the stopping criteria of numerical methods, we have used tolerance $\text{tol} = 1e - 3$.

**Remarks:**

1. From Table 4, it is observed that the SNR and SSIM by MC-based (GMRES and PGMRES) methods are quite higher than the TV-based CG method. The same comparison can be observed from Figure 7(b) to (f). So MC-based (GMRES and PGMRES) methods are generating better-denoised images.

2. From Figure 8, one can clearly observe the effectiveness of preconditioning. The number of MC-based PGMRES iterations is lesser than as compared to both MC-based GMRES and TV-based CG methods to reach the required accuracy $\text{tol} = 1e^{-3}$. The later FPIs are also having similar results. It is also observed that the TV-based CG method is showing abrupt behavior.

3. From Table 4, it is observed that the SNR and SSIM obtained by using the MC-based GMRES method are almost the same as compared with the ordinary MC-based GMRES method. But the PGMRES method is generating the same PSNR and SSIM in quite less iterations. To achieve the same PSNR and SSIM, the $P_1$ GMRES and $P_2$ GMRES methods need only 37 iterations, while the $P_3$ GMRES method needs only 65 iterations. But the GMRES method needs 70 plus iterations to get the same PSNR and SSIM. The TV-based CG method is also getting 70 plus iterations to get its litter lower SNR and SSIM. Which means that the MC-based PGMRES

| Table 4. Comparison of CG, GMRES, and PGMRES for Example 4. |
|-------------------------------------------------------------|
| Noisy SNR | CG | GMRES | $P_1$ GMRES | $P_2$ GMRES | $P_3$ GMRES |
| Noisy SSIM | 0.5264 | 0.5264 | 0.5264 | 0.5264 | 0.5264 |
| Denoised PSNR | 17.6802 | 17.6802 | 17.6840 | 17.6985 | 17.6800 |
| Denoised SSIM | 0.8333 | 0.8333 | 0.8118 | 0.8333 |
| Iterations | 70+ | 70+ | 37 | 65 | 37 |

CG: conjugate gradient; GMRES: generalized minimal residual; PGMRES: preconditioned GMRES; SNR: signal-to-noise ratio; PSNR: peak signal-to-noise ratio; SSIM: structured similarity index measure.
Figure 8. The CG, GMRES, and PGMRES convergence at FPI $m = 1$ for Example 4. The norm of the residual at each iteration is presented on the vertical axis. The blue line represents CG iterations, the orange line represents GMRES iterations, the yellow line represents $P_1$ GMRES iteration, the purple line represents $P_2$ GMRES iteration, and the green line represents $P_3$ GMRES iteration. CG: conjugate gradient; GMRES: generalized minimal residual; PGMRES: preconditioned GMRES; FPI: fixed point iteration.
Conclusion

A numerical algorithm (PGMRES) is presented to solve the primal form of MC-based nonlinear image deblurring problem. Three new circulant preconditioned matrices \( P_1, P_2, \) and \( P_3 \) are introduced. Four examples are tested by PGMRES using our new preconditioned matrices. Different kinds of images (complicated, real, synthetic, and nontexture) are tested by PGMRES using our new circulant preconditioned matrices. In Examples 3 and 4, we have compared our MC-based algorithm (PGMRES) with TV-based algorithm (CG) and observed the effectiveness of our algorithm on real and nontexture images. The convergence rates for a solution to the linear system and the norm of the residual at each iteration are presented in each example. Numerical experiments have shown the rapid convergence of the PGMRES method using new circulant preconditioners. Which means that the PGMRES method is robust, faster, and effective for an MC-based nonlinear image deblurring problem.

Acknowledgements

The authors would like to acknowledge the support and funding provided by the Deanship of Scientific Research (DSR) at KFUPM through a small business project (SB181013).

Declaration of conflicting interests

The authors declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The authors disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported by the Deanship of Scientific Research (DSR) at KFUPM through a small business project (SB181013).

ORCID iD

Shahbaz Ahmad https://orcid.org/0000-0002-9901-0924

References

1. Acar R and Vogel CR. Analysis of bounded variation penalty methods for ill-posed problems. *Inverse Probl* 1994; 10: 1217–1229.
2. Fairag F, Al-Mahdi A and Ahmad S. Two-level method for the total fractional-order variation model in image deblurring problem. *Numer Algorithms* 2020; 85: 931–950.
3. Rudin LI, Osher S and Fatemi E. Nonlinear total variation based noise removal algorithms. *Physica D Nonlinear Phenomena* 1992; 60: 259–268.
4. Vogel CR and Oman ME. Fast, robust total variation-based reconstruction of noisy, blurred images. *IEEE Trans Image Process* 1998; 7: 813–824.
5. Wang Y, Yang J, Yin W et al. A new alternating minimization algorithm for total variation image reconstruction. *SIAM J Imag Sci* 2008; 1: 248–272.
6. Chen K. Introduction to variational image-processing models and applications. *Int J Comput Math* 2013; 90: 1–8.
7. Donatelli M. A multigrid for image deblurring with Tikhonov regularization. *Numer linear Algebra Appl* 2005; 12: 715–729.
8. Faisal F and Shabbaz A. A Two-Level Method for Image Deblurring Problem. In: 2019 8th International Conference on Modeling Simulation and Applied Optimization (ICMSAO), Manama, Bahrain, 2019, pp. 1–5.
9. Fairag F, Chen K and Ahmad S. Analysis of the CCFD method for mc-based image denoising problems. *Electron Trans Numer Anal* 2021; 54: 108–127.
10. Marquina A and Osher S. Explicit algorithms for a new time dependent model based on level set motion for nonlinear deblurring and noise removal. *SIAM J Sci Comput* 2000; 22: 387–405.
11. Schmidt U, Schelten K and Roth S. Bayesian deblurring with integrated noise estimation. In: *CVPR 2011*, 2011, pp. 2625–2632. IEEE.
12. Schuler CJ, Christopher Burger H, Harmeling S et al. A machine learning approach for non-blind image deconvolution. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2013, pp. 1067–1074.
13. Su S, Delbracio M, Wang J et al. Deep video deblurring for handheld cameras. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1279–1288.
14. Sun L and Chen K. A new iterative algorithm for mean curvature-based variational image denoising. *BIT Numer Math* 2014; 54: 523–553.
15. Xu J, Chang HB and Qin J. Domain decomposition method for image deblurring. *J Comput Appl Math* 2014; 271: 401–414.
16. Yang F, Chen K, Yu B et al. A relaxed fixed point method for a mean curvature-based denoising model. *Optim Methods Softw* 2014; 29: 274–285.
17. Zhu W and Chan T. Image denoising using mean curvature of image surface. *SIAM J Imag Sci* 2012; 5: 1–32.
18. Zhu W, Tai X-C and Chan T. Augmented Lagrangian method for a mean curvature based image denoising model. *Inverse Problems Imag* 2013; 7: 1409–1432.
19. Tikhonov AN. Regularization of incorrectly posed problems. *Sov Math Dokl* 1963; 4: 1624–1627.
20. Brito-Loeza C and Chen K. Multigrid algorithm for high order denoising. *SIAM J Imag Sci* 2010; 3: 363–389.
21. Myllykoski M, Glowinski R, Karkkainen T et al. A new augmented Lagrangian approach for l^1-mean curve image denoising. *SIAM J Imaging Sci* 2015; 8: 95–125.
22. Yang F, Chen K and Yu B. Homotopy method for a mean curvature-based denoising model. *Appl Numer Math* 2012; 62: 185–200.
23. Axelsson O. *Iterative solution methods*. New York: Cambridge University Press, 1996.
24. Brito-Loeza C, Chen K and Uc-Cetina V. Image denoising using the gaussian curvature of the image surface. *Numer Methods Partial Differ Equ* 2016; 32: 1066–1089.