(2022) Computational and Applied Mathematics [Springer]

An effective algorithm for mean curvature-based image deblurring problem"

FAISAL FAIRAG*, KE CHEN †, AND SHAHBAZ AHMAD‡

Abstract. The mean curvature-based image deblurring model is widely used to preserve edges and remove staircase effect in the resulting images. However, the Euler-Lagrange equations of mean curvature model lead to solving nonlinear fourth order integro-differential equation. Furthermore the discretization of Euler-Lagrange equations produce a nonlinear ill-conditioned system which affect the convergence of the numerical algorithms like Krylov subspace methods (GMRES etc.) In this paper, we have treated the high order nonlinearity by converting the nonlinear fourth order integro-differential equation into a system of first order equations. To overcome the problem of slow convergence by GMRES method, we have introduced a new circulant peconditioned matrix. Fast convergence has shown in the numerical results by using the proposed new preconditioner. The first order error estimates are also established on a uniform rectangular mesh. The theoretical analysis is verified by showing the convergence rates in numerical examples.

Key words. Image deblurring, mean curvature, ill-posed problem, numerical analysis, precondition matrix

AMS subject classifications. 68U10, 94A08, 65N06, 65N12

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, blurrred images. One of them, ofcourse, is nonlinearty and other is the solution of the large system which arise from the discretization after linearization. The main focus of this paper to handle these two computational difficulties. The most well-known nonlinear variational image deblurring model is total variation (TV) model [1, 19, 24]. It has nice properties like edge preserving. But the main drawback of 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) [9, 22, 27, 28, 29] 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 Euler-Lagrange equations of mean curvature model lead to solving nonlinear fourth order integro-differential equation. This nonlinear high order term comes from the MC functional. Furthermore the discretization of Euler-Lagrange equations produce a nonlinear ill-conditioned system which affect the convergence of the numerical algorithms like Krylov subspace methods (GMRES etc.) The Jacobian matrix of the ill-conditioned system is having a block banded structure with large bandwidth. The mean curvature-based regularization methods are effective, but due to high nonlinearity and ill-conditioned system, robust and fast numerical solution is a crucial issue. In this manuscript, we have treated the high order nonlinearity by converting the fourth order integro-differential equation into a system of first order equations. Then we discretize this system by using cell-centered finite difference scheme along with midpoint quadrature scheme for integral term. The resulting discretized sys-

*Department of Mathematics, KFUPM, Saudi Arabia (ffairag@kfupm.edu.sa).
†Department of Mathematical Sciences, UOL, United Kingdom (k.chen@liv.ac.uk).
‡Department of Mathematics, KFUPM, Saudi Arabia (shahbazahmad@kfupm.edu.sa).

tem have a nonsymmetric structure, so GMRES method is suitable for the solution. To overcome the problem of slow convergence by GMRES method, we have introduced a new circulant peconditioner matrix. So instead of applying ordinary GMRES (Generalized Minimal Residual) method (without preconditioner) we use PGMRES (preconditioned Generalized Minimal Residual) method (with new preconditioner) for the solution of the system. Fast convergence has shown in the numerical results by using proposed new preconditioner. The first order error estimates are also established on a uniform rectangular mesh. The theoretical analysis is verified by showing the convergence rates in numerical experiments.

The contributions of the paper is as follows: (i) presents the discretization of the MC functional in terms of simple matrices; (ii) introduce a new circulant preconditioner matrix which overcomes the problem of slow convergence of GMRES method; (iii) presents a global error analysis for the numerical scheme and initiate that the error is of first-order; and, (iv) verify the theoretical result by showing the convergence rates in numerical examples. 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 for MC-based image deblurring model. The cell discretization and CCFD method are also presented in third section. In fourth section, we present the global error analysis of the method. In fifth section, we present the numerical implementation of our method. The proposed circulant preconditioner and numerical experiments are also in the fifth section. The conclusions about the proposed method is discussed in the sixth section. The appendix is the last section of the paper in which we present the supplementary material.

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

$$z = \vec{K}u + \epsilon \tag{2.1}$$

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 called the blurring operator;

$$(\vec{K}u)(x) = \int_{\Omega} k(x,y)u(y)\ dy, x \in \Omega \tag{2.2}$$

where $k(x,y) = k(x-y)$ is known as translate invariant kernel. If $\vec{K} = I$ (identity operator), then problem (2.1) is called image denoising problem. The $\vec{K}$ is a Fredholm-integral operator of first kind, so is compact. That is why the problem (2.1) becomes ill-posed [1, 23, 24]. Let Ω denotes a square in R^2. The $u \in \Omega$ is known as image intensity function. The $\mathbf{x} = (x,y)$ defines the position in Ω. Let $|\mathbf{x}| = \sqrt{x^2+y^2}$ be the Euclidean norm and $\|.\|$ is $L_2(\Omega)$ norm. The inverse problem of recovering of u from z makes (2.1) an unstable problem [1, 23, 24]. To make it stable, one remedy is to use the mean curvature (MC) regularization functional [9, 22, 27, 29],

$$J(u) = \int_{\Omega} \kappa(u)^2 \mathbf{dx} = \int_{\Omega} (\bigtriangledown . \frac{\bigtriangledown u}{|\bigtriangledown u|})^2 \mathbf{dx}. \tag{2.3}$$

Then the problem (2.1) takes the form, find u that minimizes the functional

$$T(u) = \frac{1}{2}\|\vec{K}u - z\|^2 + \frac{\alpha}{2}J(u) \tag{2.4}$$

where $\alpha > 0$ is a regularization parameter. The well-posedness of the problem (2.4) for a particular case (synthetic image denoising problem) is explained in [29]. Then the Euler-Lagrange equations of (2.4) are,

$$\vec{K}^*(\vec{K}u - z) + \alpha \bigtriangledown .[\frac{\bigtriangledown \kappa}{\sqrt{|\bigtriangledown u|^2 + \beta^2}} - \frac{\bigtriangledown \kappa . \bigtriangledown u}{(\sqrt{|\bigtriangledown u|^2 + \beta^2})^3} \bigtriangledown u] = 0 \text{ in } \Omega, \tag{2.5}$$

$$\frac{\partial u}{\partial n} = 0 \text{ in } \partial\Omega, \tag{2.6}$$

$$\kappa(u) = 0 \text{ in } \partial\Omega, \tag{2.7}$$

where $\vec{K}^*$is the adjoint operator of $\vec{K}$ and $\beta > 0$ is used to avoid non-differentiability at zero. The (2.5) is a nonlinear fourth order differential equation.

Similarly we can define corresponding mean curvature-based one-dimensional image (signal) deblurring problem. For convenience, we still denote the original image and blurry image by u and z respectively. The closed interval $I = [a, b]$ is the domain. In one-dimensional case, the MC functional have the following form;

$$\kappa(u) = (\frac{u_x}{\sqrt{u_x^2 + \beta^2}})_x \tag{2.8}$$

and the Euler-Lagrange equation are

$$\vec{K}^*(\vec{K}u - z) + \alpha\Big(\kappa_x \frac{1}{(\sqrt{u_x^2 + \beta^2})^3}\Big)_x = 0, \tag{2.9}$$

$$u_x(0) = u_x(1) = 0, \tag{2.10}$$

$$\kappa(0) = \kappa(1) = 0. \tag{2.11}$$

The mean curvature-based model not only preserve edges but also remove 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 converting the nonlinear fourth order Euler-Lagrange equation into a system of first order equations.

3. The first order nonlinear system. The equation (2.5) can be expressed as first order nonlinear system,

$$\vec{K}^*\vec{K}u + \alpha \bigtriangledown . \overrightarrow{p} - \alpha \bigtriangledown . \overrightarrow{t} = \vec{K}^* z, \tag{3.1}$$

$$-w + \bigtriangledown . \overrightarrow{v} = 0, \tag{3.2}$$

$$\sqrt{|\bigtriangledown u|^2 + \beta^2}\,\overrightarrow{v} - \bigtriangledown u = 0, \tag{3.3}$$

$$\sqrt{|\bigtriangledown u|^2 + \beta^2}\,\overrightarrow{p} - \bigtriangledown w = 0, \tag{3.4}$$

$$\sqrt{|\bigtriangledown u|^2 + \beta^2}\,\overrightarrow{t} - (\bigtriangledown w . \overrightarrow{v})\overrightarrow{v} = 0, \tag{3.5}$$

where

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

In case of one-dimensional problem, we have to introduce fewer unknowns to formulate a first order nonlinear system because (2.9) is simpler than (2.5). So from (2.9), we have the following system of four equations

$$\vec{K}^*\vec{K}u + \alpha p_x = \vec{K}^* z, \tag{3.6}$$

$$-w + v_x = 0, \tag{3.7}$$

$$\sqrt{u_x^2+\beta^2}v - u_x = 0, \tag{3.8}$$

$$(\sqrt{u_x^2+\beta^2})^3 p - w_x = 0, \tag{3.9}$$

where

$$v = \frac{u_x}{\sqrt{u_x^2+\beta^2}}, \quad w = v_x \quad \text{and} \quad p = \frac{w_x}{(\sqrt{u_x^2+\beta^2})^3}.$$

3.1. Cell discretization. For the MC-based two-dimensional image deblurring problem, the domain $\Omega = (0,1)\times(0,1)$ is partitioned by $\delta_x \times \delta_y$, [20], where

$$\delta_x : 0 = x_{1/2} < x_{3/2} < x_{5/2} < ... < x_{n_x-1/2} < x_{n_x+1/2} = 1,$$
$$\delta_y : 0 = y_{1/2} < y_{3/2} < y_{5/2} < ... < y_{n_x-1/2} < y_{n_x+1/2} = 1$$

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

$$x_i = (i-\frac{1}{2})h \quad i = 1,2,3,...,n_x,$$

$$y_j = (j-\frac{1}{2})h \quad j = 1,2,3,...,n_x$$

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,

$$x_{i\pm\frac{1}{2}} = x_i \pm \frac{h}{2} \quad i = 1,2,3,...,n_x,$$

$$y_{j\pm\frac{1}{2}} = y_j \pm \frac{h}{2} \quad j = 1,2,3,...,n_x.$$

For each $i = 1,2,...,n_x$, and $j = 1,2,...,n_x$, define

$$\Omega_{i,j} = (x_{i-1/2}, x_{i+1/2}) \times (y_{j-1/2}, y_{j+1/2}),$$

$$\Omega_{i+1/2,j} = (x_i, x_{i+1}) \times (y_{j-1/2}, y_{j+1/2}),$$

$$\Omega_{i,j+1/2} = (x_{i-1/2}, x_{i+1/2}) \times (y_j, y_{j+1}).$$

For the function $\theta(x,y)$, let $\theta_{k,l}$ denotes $\theta(x_l, y_m)$, where k and l may take values $i, i+1/2$ and $j, j+1/2$ respectively, for integers $i, j \geq 0$. For discrete functions we need values at proper discrete points, so we define

$$[d_x\theta]_{i+1/2,j} = \frac{\theta_{i+1,j}-\theta_{i,j}}{h}, \quad [D_x\theta]_{i,j} = \frac{\theta_{i+1/2,j}-\theta_{i-1/2,j}}{h},$$
$$[d_y\theta]_{i,j+1/2} = \frac{\theta_{i,j+1}-\theta_{i,j}}{h}, \quad [D_y\theta]_{i,j} = \frac{\theta_{i,ji+1/2}-\theta_{i,j-1/2}}{h}.$$

By applying midpoint quadrature approximation, we have

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

For the discretization of one dimensional image deblurring problem, one can consider above discussion in $x-$direction only.

3.2. The CCFD method. Here, we consider the cell-centered finite difference (CCFD) method for mean curvature-based image deblurring problem. The CCFD approximations $\{U_{i,j}\}$ and $\{W_{i,j}\}$ to $\{u(x_{i,j})\}$ and $\{w(x_{i,j})\}$, respectively are chosen. Denote vectors v, p and t by $v = (v^x, v^y), p = (p^x, p^y)$ and $t = (t^x, t^y)$ respectively. The CCFD approximations $\{V^x_{i+\frac{1}{2},j}\}$ and $\{V^y_{i,j+\frac{1}{2}}\}$ to $\{v^x(x_{i+\frac{1}{2},j})\}$ and $\{v^y(y_{i,j+\frac{1}{2}})\}$, respectively, are chosen. Similarly, the CCFD approximations $\{P^x_{i+\frac{1}{2},j}\}$ and $\{P^y_{i,j+\frac{1}{2}}\}$ to $\{p^x(x_{i+\frac{1}{2},j})\}$ and $\{p^y(y_{i,j+\frac{1}{2}})\}$, respectively, are chosen. And the CCFD approximations $\{T^x_{i+\frac{1}{2},j}\}$ and $\{T^y_{i,j+\frac{1}{2}}\}$ to $\{t^x(x_{i+\frac{1}{2},j})\}$ and $\{t^y(y_{i,j+\frac{1}{2}})\}$, respectively, are chosen. So we have

(3.10)

$$[K^*KU]_{i,j} + \alpha\Big([D_xP^x]_{i,j} + [D_yP^y]_{i,j}\Big) - \alpha\Big([D_xT^x]_{i,j} + [D_yT^y]_{i,j}\Big) = [K^*Z]_{i,j},$$

$$-W_{i,j} + [D_xV^x]_{i,j} + [D_yV^y]_{i,j} = 0, \tag{3.11}$$

$$\sqrt{[d_xU]^2_{i+\frac{1}{2},j} + \beta^2}V^x_{i+\frac{1}{2},j} - [d_xU]_{i+\frac{1}{2},j} = 0, \tag{3.12}$$

$$\sqrt{[d_yU]^2_{i,j+\frac{1}{2}} + \beta^2}V^y_{i,j+\frac{1}{2}} - [d_yU]_{i,j+\frac{1}{2}} = 0, \tag{3.13}$$

$$\sqrt{[d_xU]^2_{i+\frac{1}{2},j} + \beta^2}P^x_{i+\frac{1}{2},j} - [d_xW]_{i+\frac{1}{2},j} = 0, \tag{3.14}$$

$$\sqrt{[d_yU]^2_{i,j+\frac{1}{2}} + \beta^2}P^y_{i,j+\frac{1}{2}} - [d_yW]_{i,j+\frac{1}{2}} = 0, \tag{3.15}$$

$$\sqrt{[d_xU]^2_{i+\frac{1}{2},j} + \beta^2}T^x_{i+\frac{1}{2},j} - \Big([d_xW]_{i+\frac{1}{2},j}V^x_{i+\frac{1}{2},j}\Big)V^x_{i+\frac{1}{2},j} = 0, \tag{3.16}$$

$$\sqrt{[d_yU]^2_{i,j+\frac{1}{2}} + \beta^2}T^y_{i,j+\frac{1}{2}} - \Big([d_yW]_{i,j+\frac{1}{2}}V^y_{i,j+\frac{1}{2}}\Big)V^y_{i,j+\frac{1}{2}} = 0. \tag{3.17}$$

According to the lexicographical ordering of the unknowns,

$$U = [U_{11} \quad U_{12} \quad ... \quad U_{n_xn_x}]^t, \quad W = [W_{11} \quad W_{12} \quad ... \quad W_{n_xn_x}]^t,$$

$$V = [V^x_{11} \quad V^x_{12} \quad ... \quad V^x_{n_x-1n_x-1} \qquad V^y_{11} \quad V^y_{12} \quad ... \quad V^y_{n_x-1n_x-1}]^t,$$

$$P = [P^x_{11} \quad P^x_{12} \quad ... \quad P^x_{n_x-1n_x-1} \qquad P^y_{11} \quad P^y_{12} \quad ... \quad P^y_{n_x-1n_x-1}]^t,$$

$$\text{and} \quad T = [T^x_{11} \quad T^x_{12} \quad ... \quad T^x_{n_x-1n_x-1} \qquad T^y_{11} \quad T^y_{12} \quad ... \quad T^y_{n_x-1n_x-1}]^t.$$

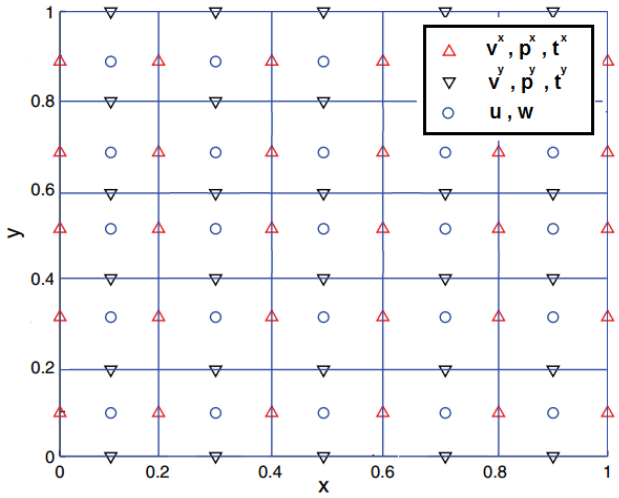


FIG. 1. *Location of variables on* 5×5 *CCFD grid.*

So from (3.10)-(3.17) we have the following matrix system,

$$K_h^* K_h U - \alpha A_h W + \alpha B_h^* P - \alpha B_h^* T = K_h^* Z, \tag{3.18}$$

$$-I_h W + B_h^* V = O, \tag{3.19}$$

$$D_h V + B_h U = O. \tag{3.20}$$

$$D_h P + B_h W = O, \tag{3.21}$$

$$D_h T - C_h V = O. \tag{3.22}$$

Here K_h, A_h and I_h are matrices of size $n_x^2 \times n_x^2$, and B_h is a matrix of size $2n_x(n_x - 1) \times n_x^2$. C_h and D_h are matrices 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 $K_h^* K_h$ is symmetric positive semidefinite. The matrix K_h is block Toeplitz with Toeplitz blocks (BTTB) matrix. 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 the identity matrix $\tilde{I}$ is size $n_x \times n_x$. 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

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

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

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

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

is a diagonal matrix and its entries are obtained by the discretization of the expression $(\bigtriangledown w.\overrightarrow{v})$. 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 that are obtained by the discretization of the expression $\sqrt{|\bigtriangledown u|^2 + \beta^2}$. The matrix D_h has the following structure,

$$D_h(U) = \begin{bmatrix} D^x & 0 \\ 0 & D^y \end{bmatrix}$$

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 Ω_{ij}, the values of the all unknowns are not available, so one can use average operators to calculate their values.

Now if we eliminate W, V, P and T from (3.18)-(3.22), then we have the following primal system,

$$(K_h^* K_h + \alpha L_h(U))U = K_h^* Z, \tag{3.23}$$

where

$$L_h = (B_h^* D_h^{-1} B_h)^2 + A_h(B_h^* D_h^{-1} B_h) + B_h^* D_h^{-1} C_h D_h^{-1} B_h. \tag{3.24}$$

According to the lexicographical ordering of the unknowns, L_h is block pentadiagonal. The diagonal blocks are pentadiagonal matrices, while the off-diagonal blocks, just below and above the main diagonal blocks, are tridiagonal matrices.

As we mentioned above, one key problem in presenting the method is to give a proper approximation to the nonlinear mean curvature functional. A number of different approximations to mean curvature functional can be find in [3, 29, 30]. Similar to the above approximation of the nonlinear mean curvature functional for image denoising problem can be found in [12] in which comparison to other methods is also presented.

4. Global Error. In this section we find the global error for MC-based image deblurring problem. For this we follow the error analysis procedure of Rui and Pan [20] who considered a Darcy−Forchheimer model using a block-centered finite difference method. For simplicity, we consider one-dimensional MC-based image deblurring problem (3.6)-(3.9). The cell-centered finite difference approximations $\{U_i\}, \{W_i\}, \{V_{i+\frac{1}{2}}\}$ and $\{P_{i+\frac{1}{2}}\}$ to $\{u(x_i)\}, \{w(x_i)\}, \{v(x_{i+\frac{1}{2}})\}$ and $\{p(x_{i+\frac{1}{2}})\}$, respectively, are chosen so that

$$[K^*KU]_i + \alpha[D_xP]_i = [K^*Z]_i, \tag{4.1}$$

$$-W_i + [D_xV]_i = 0, \tag{4.2}$$

$$\sqrt{[d_xU]^2_{i+\frac{1}{2}} + \beta^2}V_{i+\frac{1}{2}} - [d_xU]_{i+\frac{1}{2}} = 0, \tag{4.3}$$

$$(\sqrt{[d_xU]^2_{i+\frac{1}{2}} + \beta^2})^3 P_{i+\frac{1}{2}} - [d_xW]_{i+\frac{1}{2}} = 0, \tag{4.4}$$

with $V_{\frac{1}{2}} = V_{n+\frac{1}{2}} = 0$ and $W_{\frac{1}{2}} = W_{n+\frac{1}{2}} = 0$.

Theorem 4.1. *Let u, w, v and p be the solutions of the system (3.6)-(3.9) on mesh Ω_h and U, W, V and P be the computed solutions of the system (4.1)-(4.4) on mesh Ω_h. Then there exists a positive constant C independent of h such that for $i = 1, 2, ..., n$*

$$|K^*K(u_i - U_i)| \leq Ch.$$

Proof. One can notice that by putting $K = I$, in (3.6)-(3.9) and (4.1)-(4.4), we get a similar one-dimensional MC-based signal denoising problem and its descritized system, respectively, because image denoising is a special case of image deblurring model. In [12], Faisal et al, has proved the above result for the case of MC-based signal denoising problem. So here what's remain to be prove is that the presence of deblurring operator $\vec{K}$ in MC-based image deblurring model does not effect the order of global error.

First discretize the interval $[0, 1]$ using $n = n_x$ distinct points. So $h = \frac{1}{n}$.

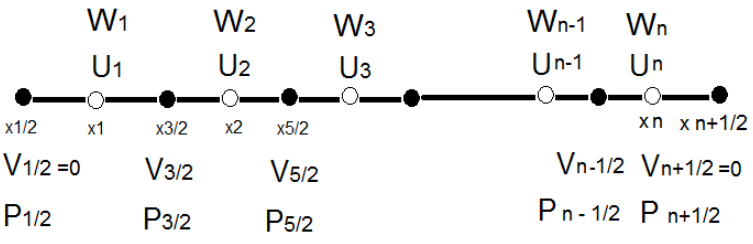


FIG. 2. *Discretization of one-dimensional MC-based image deblurring problem*

At $x = x_i, i = 1, 2, ..., (n-1)$

From (3.6) by integration, we have

$$p_{i+\frac{1}{2}} = p_{i-\frac{1}{2}} + \frac{1}{\alpha}\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} (\vec{K}^*\vec{K}u - \vec{K}^*z)dx.$$

Now by applying recursively $i-1$ times, we have

$$p_{i+\frac{1}{2}} = p_{\frac{1}{2}} + \sum_{l=1}^{i}\frac{1}{\alpha}\int_{x_{l-\frac{1}{2}}}^{x_{l+\frac{1}{2}}} (\vec{K}^*\vec{K}u - \vec{K}^*z)dx. \tag{4.5}$$

Now from (4.1), we have

$$P_{i+\frac{1}{2}} = P_{i-\frac{1}{2}} + \frac{1}{\alpha}\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} ([K^*KU]_i - [K^*Z]_i)dx.$$

Now by applying recursively $i-2$ times, we have

$$P_{i+\frac{1}{2}} = P_{\frac{1}{2}} + \sum_{l=1}^{i}\frac{1}{\alpha}\int_{x_{l-\frac{1}{2}}}^{x_{l+\frac{1}{2}}} ([K^*KU]_l - [K^*Z]_l)dx. \tag{4.6}$$

By subtracting (4.6) from (4.5), we have

$$p_{i+\frac{1}{2}} - P_{i+\frac{1}{2}} = p_{\frac{1}{2}} - P_{\frac{1}{2}} + \sum_{l=1}^{i}\frac{1}{\alpha}\int_{x_{l-\frac{1}{2}}}^{x_{l+\frac{1}{2}}} (\vec{K}^*\vec{K}u - \vec{K}^*z)dx - \sum_{l=1}^{i}\frac{1}{\alpha}\int_{x_{l-\frac{1}{2}}}^{x_{l+\frac{1}{2}}} ([K^*KU]_l - [K^*Z]_l)dx.$$

$$\Rightarrow e^p_{i+\frac{1}{2}} = e^p_{\frac{1}{2}} + \sum_{l=1}^{i}\frac{1}{\alpha}\int_{x_{l-\frac{1}{2}}}^{x_{l+\frac{1}{2}}} (\vec{K}^*\vec{K}u - [K^*KU]_l)dx - \sum_{l=1}^{i}\frac{1}{\alpha}\int_{x_{l-\frac{1}{2}}}^{x_{l+\frac{1}{2}}} (\vec{K}^*z - [K^*Z]_l)dx.$$

where $e^p_{i+\frac{1}{2}} = p_{i+\frac{1}{2}} - P_{i+\frac{1}{2}}, i = 0, 2, ..., n-1$. Now by mid-point quadrature rule, we have

$$e^p_{i+\frac{1}{2}} = e^p_{\frac{1}{2}} + \sum_{l=1}^{i}\Big(\frac{1}{\alpha}h(\vec{K}^*\vec{K}u(x_l) - [K^*KU]_l) + O(h^3)\Big) - \sum_{l=1}^{i}\Big(\frac{1}{\alpha}h(\vec{K}^*z(x_l) - [K^*Z]_l) + O(h^3)\Big).$$

Since $\vec{K}^*z(x_l) - [K^*Z]_l = 0$, so

$$e^p_{i+\frac{1}{2}} = e^p_{\frac{1}{2}} + \sum_{l=1}^{i}\frac{1}{\alpha}h(\vec{K}^*\vec{K}u(x_l) - [K^*KU]_l) + O(h^2).$$

By using Appendix I,

$$e^p_{i+\frac{1}{2}} = C_{p_{\frac{1}{2}}}h^2 + O(h^4) + \sum_{l=1}^{i}\frac{1}{\alpha}h(\vec{K}^*\vec{K}u(x_l) - [K^*KU]_l) + O(h^2).$$

$$\Rightarrow e^p_{i+\frac{1}{2}} = \sum_{l=1}^{i}\frac{1}{\alpha}h(\vec{K}^*\vec{K}u(x_l) - [K^*KU]_l) + O(h^2). \tag{4.7}$$

By using Appendix II,

$$e^p_{i+\frac{1}{2}} = \sum_{l=1}^{i} \frac{h}{\alpha}([K^*K\tilde{U}]_l - [K^*KU]_l) + O(h^2).$$

$$\Rightarrow e^p_{i+\frac{1}{2}} = \sum_{l=1}^{i} \frac{h}{\alpha}[K^*K(\tilde{U}-U)]_l + O(h^2).$$

Hence, for $i = 1, 2, ..., n-1$

$$e^p_{i+\frac{1}{2}} = \sum_{l=1}^{i} \frac{h}{\alpha} K^*Ke^u_l + O(h^2). \tag{4.8}$$

where $K^*Ke^u_l = [K^*K(\tilde{U}-U)]_l$.

At $x = x_n$

From (3.6) by integration, we have

$$p_{n+\frac{1}{2}} - p_{n-\frac{1}{2}} = \frac{1}{\alpha}\int_{x_{n-\frac{1}{2}}}^{x_{n+\frac{1}{2}}} (\vec{K}^*\vec{K}u - \vec{K}^*z)dx.$$

So

$$p_{n-\frac{1}{2}} = p_{n+\frac{1}{2}} - \frac{1}{\alpha}\int_{x_{n-\frac{1}{2}}}^{x_{n+\frac{1}{2}}} (\vec{K}^*\vec{K}u - \vec{K}^*z)dx. \tag{4.9}$$

Now from (4.1), we have

$$P_{n+\frac{1}{2}} - P_{n-\frac{1}{2}} = \frac{1}{\alpha}\int_{x_{n-\frac{1}{2}}}^{x_{n+\frac{1}{2}}} ([K^*KU]_n - [K^*Z]_n).$$

So

$$P_{n-\frac{1}{2}} = P_{n+\frac{1}{2}} - \frac{1}{\alpha}\int_{x_{n-\frac{1}{2}}}^{x_{n+\frac{1}{2}}} ([K^*KU]_n - [K^*Z]_n). \tag{4.10}$$

By subtracting (4.10) from (4.9), we have

$$p_{n-\frac{1}{2}} - P_{n-\frac{1}{2}} = p_{n+\frac{1}{2}} - P_{n+\frac{1}{2}} - \frac{1}{\alpha}\int_{x_{n-\frac{1}{2}}}^{x_{n+\frac{1}{2}}} (\vec{K}^*\vec{K}u - \vec{K}^*z)dx + \frac{1}{\alpha}\int_{x_{n-\frac{1}{2}}}^{x_{n+\frac{1}{2}}} ([K^*KU]_n - [K^*Z]_n).$$

$$\Rightarrow e^p_{n-\frac{1}{2}} = e^p_{n+\frac{1}{2}} - \frac{1}{\alpha}\int_{x_{n-\frac{1}{2}}}^{x_{n+\frac{1}{2}}} (\vec{K}^*\vec{K}u - [K^*KU]_n)dx + \frac{1}{\alpha}\int_{x_{n-\frac{1}{2}}}^{x_{n+\frac{1}{2}}} (\vec{K}^*z - [K^*Z]_n).$$

By mid-point quadrature and using $[K^*KU]_n - [K^*Z]_n = 0$, we have

$$e^p_{n-\frac{1}{2}} = e^p_{n+\frac{1}{2}} - \frac{h}{\alpha}(\vec{K}^*\vec{K}u(x_n) - [K^*KU]_n) + O(h^3).$$

By using Appendix I,

$$e^p_{n-\frac{1}{2}} = C_{p_{n+\frac{1}{2}}} h^2 + O(h^4) - \frac{h}{\alpha}(\vec{K}^*\vec{K}u(x_n) - [K^*KU]_n) + O(h^3).$$

$$\Rightarrow e^p_{n-\frac{1}{2}} = -\frac{h}{\alpha}(\vec{K}^*\vec{K}u(x_n) - [K^*KU]_n) + O(h^2).$$

Now by using (4.7), we have

$$\sum_{l=1}^{n-1} \frac{1}{\alpha} h(\vec{K}^*\vec{K}u(x_l) - [K^*KU]_l) + O(h^2) = -\frac{h}{\alpha}(\vec{K}^*\vec{K}u(x_n) - [K^*KU]_n) + O(h^2).$$

After rearranging, we have

$$\Rightarrow \sum_{l=1}^{n} \frac{h}{\alpha}(\vec{K}^*\vec{K}u(x_l) - [K^*KU]_l) + O(h^2) = 0.$$

By Appendix II, we have

$$\sum_{l=1}^{n} \frac{h}{\alpha}([K^*K\tilde{U}]_l - [K^*KU]_l) + O(h^2) = 0$$

$$\Rightarrow \sum_{l=1}^{n} \frac{h}{\alpha}[K^*K(\tilde{U} - U)]_l + O(h^2) = 0$$

So we have

$$\sum_{l=1}^{n} \frac{h}{\alpha} K^*Ke^u_l + O(h^2) = 0 \tag{4.11}$$

where $K^*Ke^u_l = [K^*K(\tilde{U} - U)]_l$. Now let

$$E^u = [K^*Ke^u_1, K^*Ke^u_2, ..., K^*Ke^u_n]^t \quad \text{and} \quad E^p = [e^p_{\frac{3}{2}}, e^p_{\frac{5}{2}}, ..., e^p_{n-\frac{1}{2}}]^t$$

So from (4.8)-(4.11), we have the following system

$$\begin{bmatrix} \frac{h}{\alpha}A_{n\times n} & B_{n\times(n-1)} \end{bmatrix} \begin{bmatrix} E^u_{n\times 1} \\ E^p_{(n-1)\times 1} \end{bmatrix} = \tilde{T}_{n\times 1}$$

where

$$A_{n\times n} = \begin{bmatrix} 1 & 0 & 0 & & \dots & 0 \\ 1 & 1 & 0 & & \dots & 0 \\ 1 & 1 & 1 & & \dots & 0 \\ & & \ddots & \ddots & \ddots & \\ 1 & 1 & 1 & \dots & \dots & 1 \end{bmatrix},$$

$$B_{n\times(n-1)} = \begin{bmatrix} -1 & 0 & 0 & \dots & 0 \\ 0 & -1 & 0 & \dots & 0 \\ & & \ddots & \ddots & \\ & & \ddots & \ddots & \\ 0 & 0 & 0 & \dots & -1 \\ 0 & 0 & 0 & \dots & 0 \end{bmatrix} = \begin{bmatrix} -I_{(n-1)\times(n-1)} \\ O_{1\times(n-1)} \end{bmatrix},$$

$$\text{and} \quad \tilde{T}_{n\times 1} = \begin{bmatrix} O(h^2) \\ O(h^2) \\ . \\ . \\ . \\ O(h^2) \\ O(h^2) \end{bmatrix}.$$

In the above system the subscript represents the order of the indicated matrix. Now by multiplying both sides with A^{-1}, we have

$$\begin{bmatrix} \frac{h}{\alpha} I_{n\times n} & A^{-1}_{n\times n} B_{n\times(n-1)} \end{bmatrix} \begin{bmatrix} E^u_{n\times 1} \\ E^p_{(n-1)\times 1} \end{bmatrix} = A^{-1}_{n\times n} \tilde{T}_{n\times 1}$$

where

$$A^{-1}_{n\times n} = \begin{bmatrix} 1 & 0 & 0 & \dots & \dots & 0 \\ -1 & 1 & 0 & \dots & \dots & 0 \\ & & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & \ddots & \\ 0 & 0 & 0 & \dots & 1 & 0 \\ 0 & 0 & 0 & \dots & -1 & 1 \end{bmatrix}.$$

So we have

$$\begin{bmatrix} \frac{h}{\alpha} I_{n\times n} & C_{n\times(n-1)} \end{bmatrix} \begin{bmatrix} E^u_{n\times 1} \\ E^p_{(n-1)\times 1} \end{bmatrix} = A^{-1}_{n\times n} \tilde{T}_{n\times 1}$$

where

$$C_{n\times(n-1)} = A^{-1}_{n\times n} B_{n\times(n-1)} = \begin{bmatrix} -A^{-1}_{(n-1)\times(n-1)} \\ D_{1\times(n-1)} \end{bmatrix}$$

and

$$D_{1\times(n-1)} = \begin{bmatrix} 0 & 0 & 0 & \dots & 0 & 1 \end{bmatrix}.$$

Then

$$\frac{h}{\alpha} E^u_{n\times 1} + C_{n\times(n-1)} E^p_{(n-1)\times 1} = A^{-1}_{n\times n} \tilde{T}_{n\times 1}$$

$$\Rightarrow E^u_{n\times 1} = -\frac{\alpha}{h} C_{n\times(n-1)} E^p_{(n-1)\times 1} + \frac{\alpha}{h} A^{-1}_{n\times n} \tilde{T}_{n\times 1}.$$

So

$$\begin{bmatrix} K^*Ke_1^u \\ K^*Ke_2^u \\ . \\ . \\ . \\ K^*Ke_n^u \end{bmatrix} = \frac{\alpha}{h} \begin{bmatrix} e^p_{\frac{3}{2}} \\ e^p_{\frac{5}{2}} - e^p_{\frac{3}{2}} \\ . \\ . \\ . \\ e^p_{n-\frac{1}{2}} - e^p_{n-\frac{3}{2}} \\ -e^p_{n-\frac{1}{2}} \end{bmatrix} + \frac{\alpha}{h} \begin{bmatrix} O(h^2) \\ O(h^2) \\ . \\ . \\ . \\ O(h^2) \\ O(h^2) \end{bmatrix}.$$

Since, for $i = 2, 3, ..., (n-1), |p_{i+\frac{1}{2}} - P_{i+\frac{1}{2}}| \leq Ch^2$ [12], hence for $i = 1, 2, ..., n$

$$|K^*K(u_i - U_i)| \leq Ch.$$

This completes the proof.

In the literature, one can find a number of numerical methods that have been investigated for mean curvature-based nonlinear minimization problems [3, 9, 16, 22, 24, 26, 27, 29]. Since MC-based models produce a large nonlinear and ill-conditioned matrix system, almost all these numerical techniques 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 provide a robust numerical method for MC-based nonlinear image deblurring problem, now we present a new preconditioned numerical method.

5. Numerical implementation. Here we introduce the algorithms to solve the MC-based nonlinear system (3.23). First we apply a discrete version of the FPI (fixed point iteration) to (3.23) to handle the nonlinearity of MC. The approach taken here is called "lagged diffusivity" scheme [24]. 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 a following linear system;

$$(K_h^*K_h + \alpha L_h(U^m))U^{m+1} = K_h^*Z. \tag{5.1}$$

5.1. Properties. Before proceeding further, we discuss some important properties of our system (5.1).

1. The Hessian matrix $K_h^*K_h + \alpha L_h(U^m)$ is extremely large for practical application. When α is small, the Hessian matrix tends to be quite ill-conditioned. This happens because the blurring operator $\bar{K}$ eigenvalues which cluster as zero [24].
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 $\bar{K}$ has translation invariant property, which allows the use of Fast Fourier transformation (FFT) to evaluate $K_h^*K_hu$ in $O(nlogn)$ operations [24].
3. The second term $L_h(U^m)$ in the Hessian matrix is sparse but not symmetric. The $L_h(U^m)$ (3.23) consists of three terms. The first and the last term in $L_h(U^m)$ are symmetric positive semidefinite [24] but the middle term is not symmetric. Hence the system (5.1) is not symmetric positive definite.

5.2. The Preconditioner. According to the properties of our system (5.1), mentioned above, Generalized Minimal Residual (GMRES) method, is suitable for the solution of the system (5.1). Due to ill-conditioned system GMRES can be quite slow to convergence. So we use preconditioned Generalized Minimal Residual (PGMRES) method [5, 6, 7, 13, 14, 15, 21]. For effective solution, preconditioning matrix P, must be symmetric positive definite (SPD) [2, 8, 24]. Here we introduce our SPD circulant preconditioned matrix P of Strang-type [17].

$$P = \alpha \tilde{K_h^*} \tilde{K_h} + \gamma diag(L_h^s(U^m)), \tag{5.2}$$

where γ is a positive parameter and where $\tilde{K_h}$ is a circulant approximation of matrix K_h. The $diag(L_h^s(U^m))$ is a diagonal matrix whose entries are the diagonal entries of matrix $L_h^s(U^m)$. The matrix

$$L_h^s = (B_h^* D_h^{-1} B_h)^2 + B_h^* D_h^{-1} C_h D_h^{-1} B_h \tag{5.3}$$

is the SPD part (first and the third term) of $L_h(U^m)$ matrix (5.1). The PGMRES method is summarized in Algorithm 5.1.

Algorithm 5.1 The PGMRES Method

On mesh Ω_h ,

1. Choose x^0 as the initial guess
2. Compute $\tilde{r}^0 = b - Ax^0$ where $A = K_h^* K_h + \alpha L_h(U^m)$ and $b = K_h^* Z$,
3. Solve $P\ r^0 = \tilde{r}^0$
4. Let $\beta_0 = \|r^0\|$, and compute $v^{(1)} = r^0/\beta_0$
5. For $k = 1, 2, ...$ until $\beta_k < \tau\beta_0$ **do**
6. $\quad \tilde{w_0}^{(k+1)} = Av^{(k)}$
7. $\quad$ Solve $P\ w_0^{(k+1)} = \tilde{w_0}^{(k+1)}$
8. $\quad$ For $l = 1$ to k **do**
9. $\quad\quad h_{lk} = \langle {w_l}^{(k+1)}, v^{(l)} \rangle$
10. $\quad\quad {w_l}^{(k+1)} = {w_l}^{(k+1)} - h_{lk} v^{(l)}$
11. $\quad$ end **do**
12. $\quad h_{k+1,k} = w_{k+l}^{(k+1)}/h_{k+1,k}$
13. $\quad$ Compute $y^{(k)}$ such that $\beta_k = \|\beta_0 e_1 - \widehat{H}_k y^{(k)}\|$ is minimized, where $\widehat{H}_k = [h_{ij}]_{1 \le i \le k+1, 1 \le j \le k}$ and $e_1 = (1, 0, ..., 0)^T$
14. end **do**
15. $x^{(k)} = x^0 + V_k y^{(k)}$

end

While applying PGMRES to (5.1), the inversion of preconditioner matrix, will be required. Since the second term $diag(L_h^s(U^m))$ in our preconditioning matrix P is diagonal matrices, so inversion can be done easily. The inversion of the first term $\tilde{K_h^*} \tilde{K_h}$, we need less than $O(nlogn)$ floating point operations using FFTs [24].

Now, let the eigenvalues of $K_h^* K_h$ and $L_h(U^m)$ 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 $P^{-1}\bar{A}$ are

$$\theta_i = \frac{\lambda_i^K + \alpha \lambda_i^L}{\alpha + \gamma \lambda_i^{L^s}}, \tag{5.4}$$

where $\lambda_i^{L^s}$ are eigenvalues of $L_h^s(U^m)$ and $\bar{A} = K_h^* K_h + \alpha L_h(U^m)$ is the Hessian matrix of the system (5.1). One can notice that $\lambda_i^{L^s} \leq \lambda_i^L$. So we have

$$\theta_i = \frac{\lambda_i^K + \alpha\lambda_i^L}{\alpha + \gamma\lambda_i^L}, \tag{5.5}$$

So clearly $\theta_i \to \frac{\alpha}{\gamma}$ as $i \to \infty$. Hence, for $\alpha \approx \gamma$, $P^{-1}\bar{A}$ has more favourable spectrum as compared to the Hessian matrix $\bar{A}$. It can also be shown in the numerical examples that PGMRES is getting rapid convergence with small γ.

5.3. Numerical experiments. Now we present five numerical examples of MC-based image deblurring problem. Here, the value of the parameters α and β is set according to [4, 29]. In all experiments, we take the zero vector to be the initial guess. We stopped the outer iterations (PGMRES) when the residuals satisfies $\|b - Ax^k\| < 10^{-7} \parallel b \parallel$ where $x^k = (v^k, u^k)$ is the solution vector in the $k-th$ iteration. We used just only one iteration of the FPI method to linearized the non-linear term and it is stopped when the tolerance is $tol = 1e-4$. In all experiments, we have used different n_x and the resulting matrix system has n_x^2 unknowns. Then the mesh size is $h = 1/n_x$. For numerical computations, MATLAB software is used on Intel(R) Core(TM) i7-4510U CPU @ 2.00 GHz 2.60 GHz. The PSNR (Peak Signal to Noise Ratio) measure is used to measure the quality of the restored images.

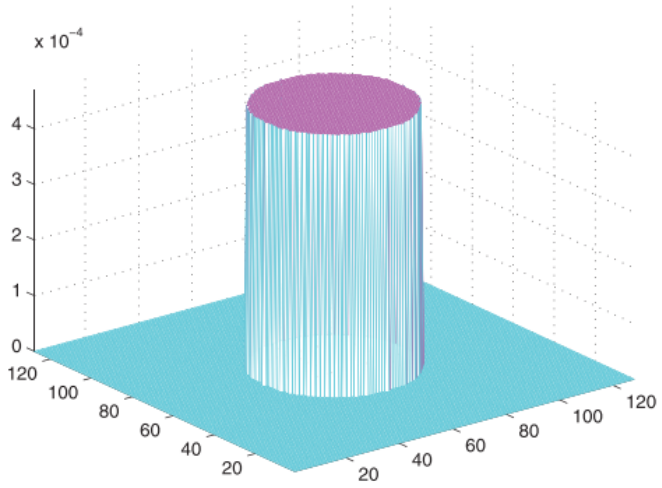
Example 1

In this example we have used the Goldhills image. The different aspects of Goldhills image are shown in Figure 5. The size of each one is 256×256. These are (a) exact image (b) blurry image (c) deblurred image without preconditioner and (d) deblurred image by using preconditioner P with $\gamma = 1e-7$. The calculation of relative residual at each iterations against different values of the parameter γ is presented in Figure 4. For numerical calculations, we have used the $ke\_gen(n_x, 300, 5)$ kernel [10, 11, 12, 18]. It is a circular *gaussian* filter of size $n_x \times n_x$ with radius $(r = 300)$ and standard deviation $(\sigma = 5)$. The $ke\_gen(120, 40, 4)$ kernel is depicted in Figure 3. The parameters $\alpha = 1e-8$ and $\beta = 1$. In Table 1, we have summarized the information of this experiment.

Remarks

1. The Figures 5(c) and 5(d) are having almost similar quality, which means that PGMRES method is generating the same quality deblurred images as by the ordinary GMRES method without preconditioning.
2. From the Figure 4, 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 ordinary GMRES (without preconditioning) to reach to the required accuracy $tol = 1e-4$. The later fixed point iterations are also having the similar results.
3. From the Figure 4, this can also be observed that PGMRES is getting rapid convergence for smaller values of γ.
4. From Table 1, it is observed that the PSNR by both methods (GMRES and PGMRES) is almost same. But PGMRES is generating same PSNR in quite less iterations. For example, to achieve the same PSNR, the PGMRES method needs only 80 iterations with $\gamma = 5e-6$. But the ordinary GMRES method needs 120 plus iterations to get the same PSNR. The number of iterations further decrease with smaller values of γ. Which means that PGMRES method is faster than the ordinary GMRES method.

5. From Table 1, it is also observed that for preconditioner P the PSNR do not get much improvement with the decrease in the value of γ. The smaller value of γ only decreases the number of iterations. So for the selection of γ, one should only care about small value of γ close to α.



(a)

FIG. 3. *The blurring kernel* $ke\_gen(120, 40, 4)$.

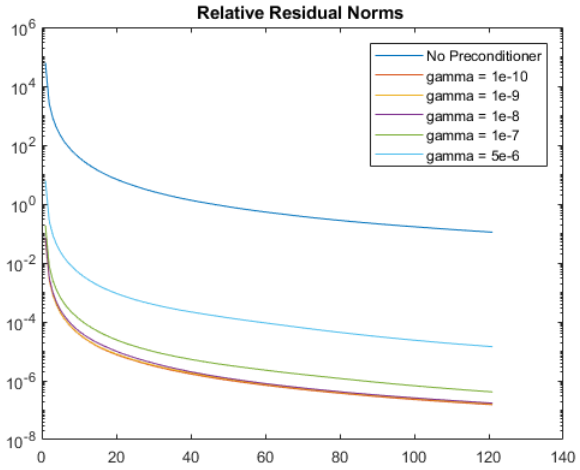
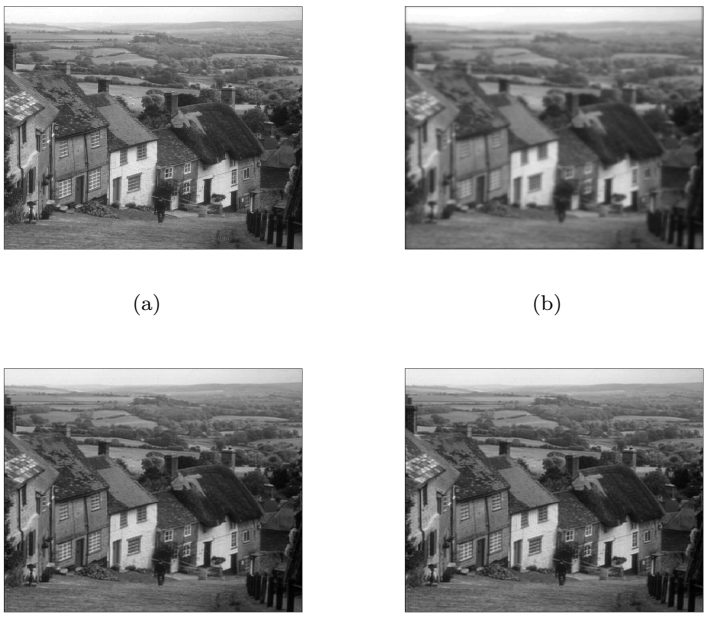


FIG. 4. *GMRES and PGMRES convergence for Example 1 at fixed point iteration* $m = 1$. *The horizontal axis represents the number of iterations. The norm of the residual at each iteration is presenting on vertical axis. Blue line represents GMRES iteations. The remaining lines represent PGMRES iteration using preconditioner* P *against different values of* γ .



(a) (b)

(c) (d)

FIG. 5. *Goldhills image: (a) exact image (b) blurry image (c) deblurred image without preconditioner and (d) deblurred image by using preconditioner P with $\gamma = 1e-7$.*

TABLE 1
Comparison of GMRES and PGMRES for Example 1

	GMRES	PGMRES				
γ	—	$1e-10$	$1e-9$	$1e-8$	$1e-7$	$5e-6$
Blurred PSNR	25.5958	25.5958	25.5958	25.5958	25.5958	25.5958
Deblurred PSNR	31.4864	29.3659	29.8812	30.7970	31.1418	31.3269
Iterations	120^{+}	10	10	11	16	80

Example 2

Here we have used the Cameraman image. This image is a complicated image, because it contains a small scale texture part (shirt) and also a large scale cartoon part (face). The different aspects of Cameraman image are shown in Figure 6. The size of each sub-figure is 512×512. These are (a) exact image (b) blurry image and (c) deblurred image by PGMRES method. In this experiment, we have used the $ke\_gen(n_x, 300, 4)$ kernel. For the analysis of convergence rate we have taken four values of n_x. These are $64, 128, 256$ and 512. Here $\beta = 1$, α is varied from $1e-5$ to $1e-7$ and γ is varied from $5e-5$ to $5e-7$. In Table 2, we have summarized the information of errors and

convergence rates of this experiment.

Remarks

1. From Figure 6, one can notice the quality of deblurred image produce by PGMRES method. Most of the blur has been removed. This means that PGMRES method generates a high quality deblurred images.
2. From the Table 2 one can observe that the relative residual norm is approximately 9×10^{-8} for all values of mesh size h. It means PGMRES method is achieving quite well accuracy.
3. The Table 2 shows that the PGMRES method have the first-order accuracy in discrete L_2 norm for all values of mesh size h. It is also observed that the error $\|K^*K(u-U)\|$ also decreases with the decrease in the mesh size h.

(a) (b) (c)

FIG. 6. *Cameraman image* : (*a*) *exact image* (*b*) *blurry image and* (*c*) *deblurred image by PGMRES method.*

TABLE 2
PGMRES calculations for Example 2

Mesh Size (h)	Blurred PSNR	Deblurred PSNR	Relative Residual	Error $\|K^*K(u-U)\|$	Rate
1/64	19.9426	48.9348	9.5696e-08	4.021e-02	–
1/128	21.4185	44.0716	9.7826e-08	1.995e-02	1.0043
1/256	20.9834	40.8261	9.9701e-08	1.060e-02	0.9123
1/512	22.1542	43.1164	9.9705e-08	4.801e-03	1.1430

Example 3
Here we have used the Peppers image. The Peppers image is a nontexture image. The different aspects of Peppers image are shown in Figure 7. The size of each subfigure is 512×512. These are (a) exact image (b) blurry image and (c) deblurred image by PGMRES method. In this experiment, we have used the $ke\_gen(n_x, 200, 4)$ kernel. For the analysis of convergence rate we have taken four values of n_x. These are $64, 128, 256$ and 512. Here $\beta = 1$, α is varied from $1e-7$ to $1e-8$ and γ is varied from $5e-7$ to $5e-8$. In Table 3, we have summarized the information of errors and convergence rates of this experiment.

Remarks

1. From Figure 7, one can notice that in the deblurred image most of the blur has been removed. This means that PGMRES method generates high quality results.
2. The Table 3 shows that the PGMRES method have the first-order accuracy in discrete L_2 norm for all values of mesh size h and the error $\|K^*K(u-U)\|$ also get decrease with the decrease in the mesh size h. The relative error is also quite low for this example too.

(a) (b) (c)

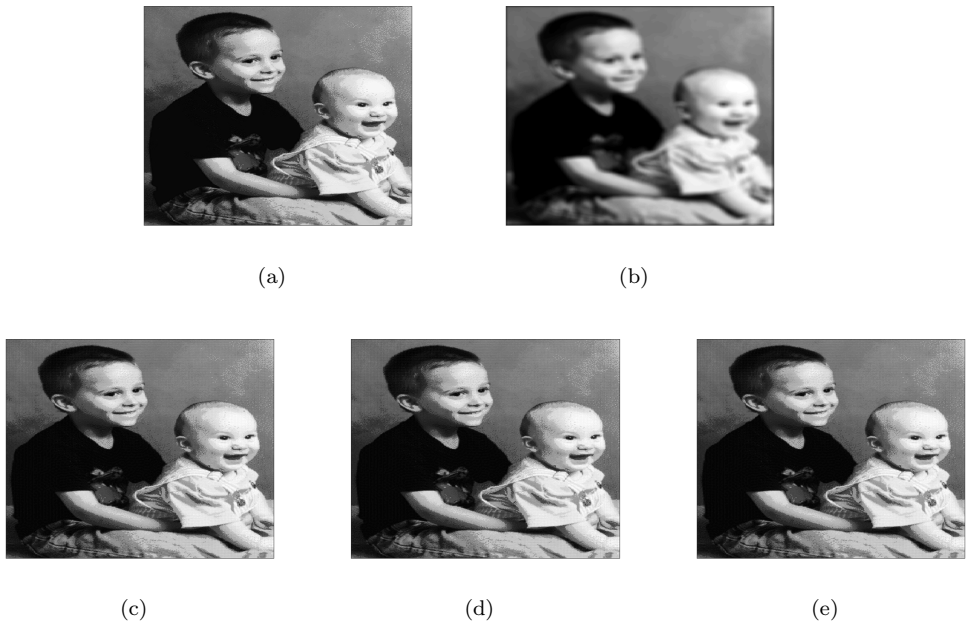
FIG. 7. *Peppers image* : (*a*) *exact image* (*b*) *blurry image and* (*c*) *deblurred image by PGMRES method.*

TABLE 3
PGMRES calculations for Example 3

Mesh Size (h)	Blurred PSNR	Deblurred PSNR	Relative Residual	Error $\|K^*K(u-U)\|$	Rate
1/64	18.8168	33.9585	9.9196e-08	2.0286e-05	–
1/128	21.3244	35.9837	9.9746e-08	3.9824e-05	0.9732
1/256	23.3415	42.3435	9.9781e-08	1.6730e-05	1.2512
1/512	23.9288	41.7332	9.9634e-08	8.5444e-05	1.2663

Example 4 In this example we have used Kids image. This image is also a complicated image, because it contains a small scale texture part (shirt) and also a large scale cartoon part (face). Here we have compared our MC based algorithm with TV (total variation) based algorithm. Since TV-based model generates a SPD matrix

system [24], so for the solution we have used CG (Conjugate Gradient) method. The different aspects of Kids image are presented in Figure 8. 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 and (e) deblurred image by PGMRES. For numerical calculations, we have used the $ke\_gen(n_x, 300, 5)$ kernel. For TV-based method we have used $\alpha = 1e-4$ and $\beta = 1$ according to [24]. For MC-based method we have used $\alpha = 1e-6, \beta = 1$ and $\gamma = 1e-5$.



(a) (b)

(c) (d) (e)

FIG. 8. *Kids image: (a) exact image (b) blurry image (c) deblurred image by CG (d) deblurred image by GMRES and (e) deblurred image by PGMRES.*

TABLE 4
Comparison of CG, GMRES and PGMRES for Example 4

	CG	GMRES	PGMRES
Blurred PSNR	22.9734	22.9734	22.9734
Deblurred PSNR	39.0931	43.2036	43.6945
$\|K^*K(u-U)\|$	3.8528e-06	6.0732e-07	7.5004e-07

Remark

From Table 4, it is observed that the PSNR by MC-based (GMRES and PGMRES) methods are little higher than TV-based CG method. Same comparison can be observed from Figures 8(c), 8(d) and 8(e). So MC-based (GMRES and PGMRES) methods are generating better quality results.

Example 5 In this example, we have compared our MC-based method with MC-based augmented lagrangian method (ALM) [25, 28, 30, 31]. In [30] Zhu et al, has presented ALM for only image denoising case (when blurring operator $\vec{K} = I$ identity operator). Here, we have extend that algorithm for image deblurring problem and presented in Algorithm 5.2. The complete detail of the ALM algorithm can be found in [30, 31].

Algorithm 5.2 The Augmented Lagrangian method for MC model

1. Initialize $u^0, q^0, \vec{p}^0, \vec{n}^0, \vec{m}^0$, and $\lambda_1^0, \vec{\lambda}_2^0, \lambda_3^0, \vec{\lambda}_4^0$.
2. For $k = 0, 1, 2, ...$: Compute $(u^k, q^k, \vec{p}^k, \vec{n}^k, \vec{m}^k)$ as an (approximate) minimizer of the augmented Lagrangian functional with the Lagrange multiplier $\lambda_1^{k-1}, \vec{\lambda}_2^{k-1}, \lambda_3^{k-1}, \vec{\lambda}_4^{k-1}$, i.e.,

$$(u^k, q^k, \vec{p}^k, \vec{n}^k, \vec{m}^k) \approx argmin\mathbf{L}(u, q, \vec{p}, \vec{n}, \vec{m}, \lambda_1^{k-1}, \vec{\lambda}_2^{k-1}, \lambda_3^{k-1}, \vec{\lambda}_4^{k-1}). \tag{5.6}$$

3. Update the Lagrangian multipliers

$$\lambda_1^k = \lambda_1^{k-1} + r_1(|\vec{p}^k| - \vec{p}^k.\vec{m}^k), \tag{5.7}$$

$$\vec{\lambda}_2^k = \vec{\lambda}_2^{k-1} + r_2(|\vec{p}^k| - < \bigtriangledown u^k, 1 >), \tag{5.8}$$

$$\lambda_3^k = \lambda_3^{k-1} + r_3(q^k - \partial n_1^k - \partial n_2^k), \tag{5.9}$$

$$\vec{\lambda}_4^k = \vec{\lambda}_4^{k-1} + r_4(\vec{n}^k - \vec{m}^k), \tag{5.10}$$

where $\vec{n} =< n_1, n_2, n_3 >$.

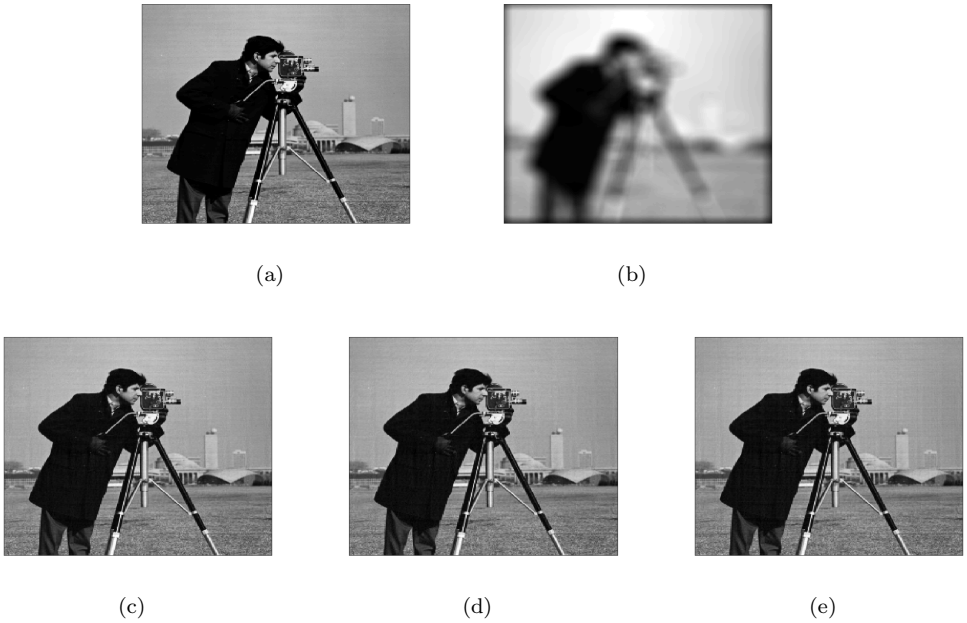
4. Measure the relative residuals and stop the iteration if they are smaller than a threshold ϵ_r.

The associated augmented Lagrangian functional in (5.6) is

$$\begin{aligned}\mathbf{L}(u, q, \vec{p}, \vec{n}, \vec{m}, \lambda_1, \vec{\lambda}_2, \lambda_3, \vec{\lambda}_4) = \frac{1}{2}\int (\vec{K}u - z)^2 + \alpha \int |q| \\ + r_1 \int (|\vec{p}| - \vec{p}.\vec{m}) + \int \lambda_1(|\vec{p}| - \vec{p}.\vec{m}) \\ + \frac{r_2}{2}\int |\vec{p} - < \bigtriangledown u, 1 >|^2 + \int \vec{\lambda}_2.(\vec{p} - < \bigtriangledown u, 1 >) \\ + \frac{r_3}{2}\int (q - \partial_x n_1 - \partial_y n_2)^2 + \int \lambda_3.(q - \partial_x n_1 - \partial_y n_2) \\ + \frac{r_4}{2}\int |\vec{n} - \vec{m}|^2 + \int \vec{\lambda}_4.(\vec{n} - \vec{m}) + \delta_R(\vec{m}),\end{aligned} \tag{5.11}$$

where r_1, r_2, r_3 and r_4 are the penalization parameters. The $\lambda_1, \lambda_3, \in R$ and $\vec{\lambda}_2, \vec{\lambda}_4 \in R^3$ are Lagrange multipliers, and $\vec{p}, \vec{n}, \vec{m} \in R^3$. All these terms except the data fidelity term are similar to the ones discussed in [30]. For the comparison we have used Cameraman image. The different aspects of Cameraman image are presented in Figure 9. The size of each subfigure is 256×256. The MC-based augmented lagrangian method use certain sets of parameters which we have used according to [30, 31]. The

parameters for ALM are $\alpha = 1e-9, r_1 = 9.5e-7, r_2 = 1e-6, r_3 = 1e-8, r_4 = 1e-5$ and $\beta = 1$. For our MC-based algorithms (GMRES and PGMRES) we have used $\alpha = 1e-9, \beta = 1$ and $\gamma = 1e-8$. For numerical calculations, we have used the $ke\_gen(n_x, 300, 10)$ kernel and for the stopping criteria of a numerical methods we have used tolerance $tol = 1e-8$.



(a) (b)

(c) (d) (e)

FIG. 9. *Cameraman image: (a) exact image (b) blurry image (c) deblurred image by ALM (d) deblurred image by GMRES and (f) deblurred image by PGMRES.*

Remarks

1. From the Figure 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. But our methods (GMRES and PGMRES) are generating same PSNR in quite less CPU-Time as compared to augmented lagrangian method (ALM). Our methods are saving quite well CPU-Time. For example, the GMRES and PGMRES methods are generating their PSNR values in 131.0810 seconds and 104.5429 seconds, respectively. But ALM is generating its PSNR in 241.1423 seconds. Which means GMRES method is saving more than 40% of CPU-Time as compared to ALM and PGMRES method is saving more than 50% of CPU-Time as compared to ALM. So we can say, our proposed MC-based methods are faster than the MC-based augmented lagrangian method.

6. Conclusion. A numerical algorithm (PGMRES) is presented to solve the primal form of mean curvature-based nonlinear image deblurring problem. A new SPD circulant preconditioner matrix is introduced. The first order error estimates are established on a uniform rectangular mesh. Five examples are tested by PGMRES using our new preconditioner matrix. Different kinds of images (Complicated, real, synthetic and nontexture) are tested by PGMRES using our new preconditioner

TABLE 5
Comparison of different methods for Example 5

Image	Blurred PSNR	Method	Deblurred PSNR	CPU-Time
Cameraman	17.8172	ALM	40.1991	241.1423
	17.8172	GMRES	38.8709	131.0810
	17.8172	PGMRES	38.8143	104.5429

matrix. We have also compared our MC based algorithm with TV (total variation) based algorithm. The comparison between our algorithms and augmented lagrangian method (ALM) is also presented. Numerical experiments show the rapid convergence of PGMRES method using new preconditioner. The first order error estimates are also established on a uniform rectangular mesh. Numerical experiments show the consistency of the convergence rates of our method with the theoretical analysis.

7. Appendix. In this section we present collection of supplementary materials. Appendix I Consider

$$e^p_{\frac{1}{2}} = p(x_{\frac{1}{2}}) - P_{\frac{1}{2}}.$$

By using (3.9) and (4.4), we have

$$e^p_{\frac{1}{2}} = \frac{1}{(\sqrt{u_x(x_{\frac{1}{2}})^2 + \beta^2})^3} w_x(x_{\frac{1}{2}}) - \frac{1}{(\sqrt{\frac{1}{h}[U_1 - U_0]^2 + \beta^2})^3} \frac{1}{h}[W_1 - W_0]$$

Since $u_x(x_{\frac{1}{2}}) = 0$ and $U_0 = U_1$, so

$$e^p_{\frac{1}{2}} = \frac{1}{\beta^3}\{w_x(x_{\frac{1}{2}}) - \frac{1}{h}[W_1 - W_0]\}$$

$$= \frac{1}{\beta^3}\{w_x(x_{\frac{1}{2}}) - \frac{1}{h}[w(x_{\frac{1}{2}} + \frac{h}{2}) - w(x_{\frac{1}{2}} - \frac{h}{2})]\}.$$

By applying Taylor's series on right side, we have

$$e^p_{\frac{1}{2}} = -\frac{h^2}{24\beta^3} w_{xxx}(x_{\frac{1}{2}}) + O(h^4).$$

Hence $e^p_{\frac{1}{2}} = C_{p_{\frac{1}{2}}} h^2 + O(h^4)$. Similarly, $e^p_{n+\frac{1}{2}} = p(x_{n+\frac{1}{2}}) - P_{n+\frac{1}{2}} = C_{p_{n+\frac{1}{2}}} h^2 + O(h^4)$.

Appendix II Consider

$$(\vec{K}u)(x) = \int_0^1 k(x,t)u(t)dt = \sum_{m=1}^{n} \int_{x_{m-\frac{1}{2}}}^{x_{m+\frac{1}{2}}} k(x,t)u(t)dt.$$

By applying mid-point quadrature rule, we have

$$(\vec{K}u)(x) = \sum_{m=1}^{n} hk(x, x_m)u(x_m) + \frac{h^3}{24}\frac{\partial^2}{\partial t^2}[k(x,t)u(t)]_{t=\xi_m}$$

where $\xi_m \in (x_{m-\frac{1}{2}}, x_{m+\frac{1}{2}})$. So for $x = x_i, i = 1, 2, ..., n$,

$$(\vec{K}u)(x_i) = \sum_{m=1}^{n} hk(x_i, x_m)u(x_m) + \frac{h^3}{24}\sum_{m=1}^{n}\frac{\partial^2}{\partial t^2}[k(x_i,t)u(t)]_{t=\xi_m}$$

where $\xi_m \in (x_{m-\frac{1}{2}}, x_{m+\frac{1}{2}})$. This can be written as

$$(\vec{K}u)(x) = \sum_{m=1}^{n} hk(x - x_m)u(x_m) + \frac{h^3}{24}\sum_{m=1}^{n} H_{tt}(x, \xi_m) \tag{7.1}$$

where $\xi_m \in (x_{m-\frac{1}{2}}, x_{m+\frac{1}{2}})$ and $H(x,t) = k(x,t)u(t)$. Similarly, we have, for $x = x_j, j = 1, 2, ..., n$

$$(\vec{K}^*u)(x) = \sum_{j=1}^{n} hk(x_j - x)u(x_j) + \frac{h^3}{24}\sum_{j=1}^{n} G_{tt}(\xi_j, x) \tag{7.2}$$

where $\xi_j \in (x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}})$ and $G(t,x) = k(t,x)u(t)$. So by using 7.1 and 7.2, we have, for $x = x_i, i = 1, 2, ..., n$

$$(\vec{K}^*\vec{K}u)(x) = \sum_{j=1}^{n} hk(x_j - x)(\vec{K}u)(x_j) + \frac{h^3}{24}\sum_{j=1}^{n} G_{tt}(\xi_j, x)$$

where $\xi_j \in (x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}})$ and $G(t,x) = k(t,x)(\vec{K}u)(t)$.

$$\Rightarrow (\vec{K}^*\vec{K}u)(x) = \sum_{j=1}^{n} hk(x_j - x)[\sum_{m=1}^{n} hk(x_j - x_m)u(x_m) + \frac{h^3}{24}\sum_{m=1}^{n} H_{tt}(x_j, \xi_m)] + \frac{h^3}{24}\sum_{j=1}^{n} G_{tt}(\xi_j, x)$$

where $\xi_j \in (x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}})$, $\xi_m \in (x_{m-\frac{1}{2}}, x_{m+\frac{1}{2}})$, $H(x,t) = k(x,t)u(t)$ and $G(t,x) = k(t,x)(\vec{K}u)(t)$.

$$\Rightarrow (\vec{K}^*\vec{K}u)(x) = \sum_{j=1}^{n}\sum_{m=1}^{n} h^2 k(x_j - x)k(x_j - x_m)u(x_m)$$

$$+\frac{h^4}{24}\sum_{j=1}^{n} k(x_j - x)\sum_{m=1}^{n} H_{tt}(x_j, \xi_m) + O(h^5), \quad \text{for} \quad x = x_i, i = 1, 2, ..., n.$$

So we have the following matrix system

$$\begin{bmatrix} (\vec{K}^*\vec{K}u)(x_1) \\ (\vec{K}^*\vec{K}u)(x_2) \\ \cdot \\ \cdot \\ \cdot \\ (\vec{K}^*\vec{K}u)(x_n) \end{bmatrix} = h^2$$

$$\begin{bmatrix} k(0) & k(h) & \dots & k((n-1)h) \\ k(-h) & k(0) & \dots & k((n-2)h) \\ & & \ddots & \\ k(-(n-1)h) & k(-(n-2)h) & \dots & k(0) \end{bmatrix} \begin{bmatrix} k(0) & k(-h) & \dots & k(-(n-1)h) \\ k(h) & k(0) & \dots & k(-(n-2)h) \\ & & \ddots & \\ k((n-1)h) & k((n-2)h) & \dots & k(0) \end{bmatrix} \begin{bmatrix} u(x_1) \\ u(x_2) \\ . \\ . \\ . \\ u(x_n) \end{bmatrix}$$
$$+\frac{h^4}{24} \begin{bmatrix} \sum_{j=1}^n k(x_j - x_1) \sum_{m=1}^n H_{tt}(x_j, \xi_m) + O(h) \\ \sum_{j=1}^n k(x_j - x_2) \sum_{m=1}^n H_{tt}(x_j, \xi_m) + O(h) \\ . \\ . \\ . \\ \sum_{j=1}^n k(x_j - x_n) \sum_{m=1}^n H_{tt}(x_j, \xi_m) + O(h) \end{bmatrix},$$

where $k(x_i - x_n) = k((n-i)h)$ for $i = 1, 2, ..., n$. This can be written as

$$\begin{bmatrix} (\vec{K}^* \vec{K} u)(x_1) \\ (\vec{K}^* \vec{K} u)(x_2) \\ . \\ . \\ . \\ (\vec{K}^* \vec{K} u)(x_n) \end{bmatrix} = K^*_{n\times n} K_{n\times n} \tilde{U}_{n\times 1} + \frac{h^4}{24} \xi^{K^* K U}_{n\times 1}.$$

So for $i = 1, 2, ..., n$

$$(\vec{K}^* \vec{K} u)(x_i) = K^* K \tilde{U}_i + O(h^5).$$

Acknowledgments

The first and the third author would like to acknowledge the DSR (Deanship of Scientific Research) at KFUPM for funding this work through small business project ($SB181013$).

REFERENCES

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

[2] M. Benzi and G. H. Golub, *A preconditioner for generalized saddle point problems*, SIAM Journal on Matrix Analysis and Applications, 26 (2004), pp. 20–41.

[3] C. Brito-Loeza and K. Chen, *Multigrid algorithm for high order denoising*, SIAM Journal on Imaging Sciences, 3 (2010), pp. 363–389.

[4] C. Brito-Loeza, K. Chen, and V. Uc-Cetina, *Image denoising using the gaussian curvature of the image surface*, Numerical Methods for Partial Differential Equations, 32 (2016), pp. 1066–1089.

[5] R. H. Chan, *Toeplitz preconditioners for toeplitz systems with nonnegative generating functions*, IMA journal of numerical analysis, 11 (1991), pp. 333–345.

[6] R. H. Chan and K.-P. Ng, *Toeplitz preconditioners for hermitian toeplitz systems*, Linear algebra and its applications, 190 (1993), pp. 181–208.

[7] T. F. Chan, *An optimal circulant preconditioner for toeplitz systems*, SIAM journal on scientific and statistical computing, 9 (1988), pp. 766–771.

[8] C. Chen and C. Ma, *A generalized shift-splitting preconditioner for saddle point problems*, Applied Mathematics Letters, 43 (2015), pp. 49–55.

[9] K. Chen, *Introduction to variational image-processing models and applications*, International Journal of Computer Mathematics, 90 (2013), pp. 1–8.

[10] K. Chen, F. Fairag, and A. Al-Mahdi, *Preconditioning techniques for an image deblurring problem*, Numerical Linear Algebra with Applications, 23 (2016), pp. 570–584.

[11] F. Fairag and S. Ahmad, *A two-level method for image deblurring problem*, in 2019 8th International Conference on Modeling Simulation and Applied Optimization (ICMSAO), IEEE, 2019, pp. 1–5.

[12] F. Fairag, K. Chen, and S. Ahmad, *Analysis of the ccfd method for mc-based image denoising problems*, Electronic Transactions on Numerical Analysis, 54 (2021), pp. 108–127.
[13] F.-R. Lin, *Preconditioners for block toeplitz systems based on circulant preconditioners*, Numerical Algorithms, 26 (2001), pp. 365–379.
[14] F.-R. Lin and W.-K. Ching, *Inverse toeplitz preconditioners for hermitian toeplitz systems*, Numerical linear algebra with applications, 12 (2005), pp. 221–229.
[15] F.-R. Lin and C.-X. Wang, *Bttb preconditioners for bttb systems*, Numerical Algorithms, 60 (2012), pp. 153–167.
[16] M. Myllykoski, R. Glowinski, T. Karkkainen, and T. Rossi, *A new augmented lagrangian approach for lˆ1-mean curvature image denoising*, SIAM Journal on Imaging Sciences, 8 (2015), pp. 95–125.
[17] M. K. Ng, *Iterative methods for Toeplitz systems*, London, U.K.: Oxford University Press, 2004.
[18] K. L. Riley, *Two-level preconditioners for regularized ill-posed problems*, PhD Thesis, Montana State University, 1999.
[19] L. I. Rudin, S. Osher, and E. Fatemi, *Nonlinear total variation based noise removal algorithms*, Physica D: Nonlinear Phenomena, 60 (1992), pp. 259–268.
[20] H. Rui and H. Pan, *A block-centered finite difference method for the Darcy-Forchheimer model*, SIAM Journal on Numerical Analysis, 5 (2012), pp. 2612–2631.
[21] D. K. Salkuyeh, M. Masoudi, and D. Hezari, *On the generalized shift-splitting preconditioner for saddle point problems*, Applied Mathematics Letters, 48 (2015), pp. 55–61.
[22] L. Sun and K. Chen, *A new iterative algorithm for mean curvature-based variational image denoising*, BIT Numerical Mathematics, 54 (2014), pp. 523–553.
[23] A. N. Tikhonov, *Regularization of incorrectly posed problems*, in Soviet Math. Dokl, vol. 4, 1963, pp. 1624–1627.
[24] C. R. Vogel and M. E. Oman, *Fast, robust total variation-based reconstruction of noisy, blurred images*, IEEE Transactions on Image Processing, 7 (1998), pp. 813–824.
[25] C. Wu and X.-C. Tai, *Augmented lagrangian method, dual methods, and split bregman iteration for rof, vectorial tv, and high order models*, SIAM Journal on Imaging Sciences, 3 (2010), pp. 300–339.
[26] F. Yang, K. Chen, and B. Yu, *Homotopy method for a mean curvature-based denoising model*, Applied Numerical Mathematics, 62 (2012), pp. 185–200.
[27] F. Yang, K. Chen, B. Yu, and D. Fang, *A relaxed fixed point method for a mean curvature-based denoising model*, Optimization Methods and Software, 29 (2014), pp. 274–285.
[28] J. Zhang, C. Deng, Y. Shi, S. Wang, and Y. Zhu, *A fast linearised augmented lagrangian method for a mean curvature based model*, East Asian Journal on Applied Mathematics, 8 (2018), pp. 463–476.
[29] W. Zhu and T. Chan, *Image denoising using mean curvature of image surface*, SIAM Journal on Imaging Sciences, 5 (2012), pp. 1–32.
[30] W. Zhu, X.-C. Tai, and T. Chan, *Augmented lagrangian method for a mean curvature based image denoising model*, Inverse Problems and Imaging, 7 (2013), pp. 1409–1432.
[31] W. Zhu, X.-C. Tai, and 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, pp. 104–118.